



***Facultad
de
Ciencias***

**DESARROLLO DE UNA HERRAMIENTA DE
CLONADO DE EQUIPOS
(DEVELOPMENT OF A COMPUTER CLONING
TOOL)**

**Trabajo de Fin de Grado
para acceder al**

GRADO EN INGENIERÍA INFORMÁTICA

Autor: Daniel Peña Cuadrado

Director: Esteban Stafford Fernández

Julio – 2021

ÍNDICE

Agradecimientos	4
Resumen	5
Abstract	6
1. Introducción	7
1.1 Motivación.....	7
1.2 Objetivo	8
2. Herramientas y servicios utilizados	9
2.1 Herramientas	9
2.1.1 VirtualBox	9
2.1.2 Live-CDs.....	9
2.1.3 Programación en shell	10
2.1.4 Dialog	10
2.1.5 UDP-Sender	11
2.2 Servicios	11
2.2.1 Dhcpd	11
2.2.2 Tftpd	12
2.2.3 Syslinux	15
2.2.4 Secuencia de arranque Linux.....	15
3. Arranque por red de System Rescue CD	17
3.1 Ficheros necesarios.....	17
3.2 Servicio PXE/TFTP en SRCD versión 5.....	19
4. Ingeniería software del proyecto	21
4.1 Metodología de trabajo usada.....	21
4.2 Requisitos funcionales	21
4.3 Requisitos no funcionales	22
4.4 Requisitos hardware y software	22
4.5 Planificación.....	23
5. Desarrollo del script	24
5.1 Script de clonado	25
5.1.1 Comunicación servidor-clientes (UDP-Sender)	26
5.1.2 Clonado de particiones de disco seleccionadas	27
5.2 Interfaz gráfica de usuario (GUI)	28
6. Evaluación y pruebas	33
6.1 Prueba de servicio dhcpd.....	33
6.2 Prueba de servicio tftpd/pxe.....	34
6.3 Modificación de imagen de System Rescue CD v5.3	35
6.4 Pruebas herramienta en entorno virtual	36
6.4.1 Pruebas desde parte servidor	37
6.4.2 Pruebas desde clientes.....	39
7. Conclusiones y trabajos futuros	42
7.1 Caso práctico de uso	42
7.2 Conclusiones	43
7.3 Trabajos futuros	44
8. Referencias bibliográficas	46

Índice de figuras

Figura 1. Diagrama de flujo simplificado PXE	12
Figura 2. Tipos de paquetes TFTP.....	13
Figura 3. Secuencia arranque por red.....	17
Figura 4. Fichero "Default" arranque por red SystemRescueCD	19
Figura 5. Gestor de red tráfico SystemRescueCD	20
Figura 6. Arranque servicio PXE SystemRescueCD	20
Figura 7. Requisitos funcionales	22
Figura 8. Requisitos no funcionales	22
Figura 9. Requisitos hardware	23
Figura 10. Programa principal (main).....	24
Figura 11. Opciones función clonar (GUI)	25
Figura 12. Opciones función clonar (ejemplo, manual)	25
Figura 13. Generación y ejecución script master y esclavo.....	26
Figura 14. Comunicación cliente-servidor con UDP-Sender.....	27
Figura 15. Diferenciación opciones y tipos de particiones	27
Figura 16. Consola SRCD	29
Figura 17. Pantalla principal GUI	29
Figura 18. Información discos duros GUI.....	29
Figura 19. Elegir disco duro GUI.....	30
Figura 20. Elegir opciones GUI	30
Figura 21. Elegir partición grub GUI.....	30
Figura 22. Ventana de confirmación GUI	31
Figura 23. Código elegir disco, Dialog - GUI	31
Figura 24. Estado servicio dhcpd.....	33
Figura 25. Arranque DHCP cliente1	33
Figura 26. Arranque DHCP cliente2.....	33
Figura 27. Log servicio dhcpd en servidor DHCP	34
Figura 28. Estado servicio tftpd.....	34
Figura 29. Carga sistema de ficheros SRCD por TFTP.....	35
Figura 30. Ficheros imagen LiveCD SRCD	35
Figura 31. Código script "script_provisionado"	36
Figura 32. Código "test" mode	37
Figura 33. Ejecución "script_provisionado"	37
Figura 34. Confirmación opciones prueba.....	37
Figura 35. Ejecución herramienta en servidor	38
Figura 36. Logging servidor	39
Figura 37. Ejecución autorun0 en cliente1	39
Figura 38. Ejecución autorun0 en cliente2	40
Figura 39. Comunicación cliente1	40
Figura 40. Comunicación cliente2	40
Figura 41. Transferencia tabla de particiones	40
Figura 42. Transferencia partición NTFS	41
Figura 43. Transferencia partición Linux	41
Figura 44. Tabla comparativa beneficios.....	43
Figura 45. Ejemplo ventana de progreso.....	43

Agradecimientos

Me gustaría aprovechar estos primeros párrafos como agradecimiento a todas aquellas personas que han estado ahí durante los últimos años, profesores, compañeros de trabajo, familia, pareja y amigos de toda la vida. No han sido años fáciles, compaginar estos últimos años de trabajo y universidad me ha resultado realmente difícil, pero aquí estoy, terminando una de las épocas necesarias en la vida. Los conocimientos del grado en Ingeniería Informática junto a la experiencia laboral, de los últimos años en este sector, me han aportado unas habilidades que, sin lugar a duda, me van a acompañar durante el largo trayecto profesional que me queda por delante.

Primeramente, darle las gracias a mi familia, sobre todo a mis padres, por creer en mí, por apoyarme en todo momento, por animarme con la frase “Nadie es más que nadie, querer es poder”, por inculcarme desde pequeño la necesidad de estudiar, por fomentar en mí el autoaprendizaje y que con trabajo todo es posible. Por último, agradecerles el gran desembolso económico que estos estudios han supuesto, en definitiva, sin mis padres estos estudios no hubiesen sido posibles.

Darle las gracias a mi pareja, Lucía, por el apoyo aportado en los malos momentos, por animarme a continuar hacia delante bajo cualquier circunstancia. También debo agradecer el interés que ha tenido en cuestiones tecnológicas en el ámbito de la educación como Maestra o en el ámbito doméstico.

Gracias a mis compañeros de trabajo de IT en SEG Automotive Spain, porque hemos pasado juntos mucho tiempo durante los últimos años, por todas las cosas que he aprendido de vosotros, cosas del sector tecnológico y cosas del ámbito personal, porque no hay mejor escuela que haberse dedicado a esto 35 años de tu vida.

A mis amigos de toda la vida, César, Miguel, Álvaro, por escuchar mis problemas e intentar comprenderme. Por darme los mejores consejos y sacarme siempre una sonrisa.

Agradecido también de todos los profesores que me han dado clase, porque de todos ellos he aprendido algo. Quisiera darles especialmente las gracias a aquellos profesores que han sembrado e incentivado mi curiosidad en el mundo de la informática, por ejemplo, a ti, Esteban. Te conozco de un par de asignaturas, pero por tu trabajo y carisma te considero un referente en cuanto a lo que me gustaría dedicarme en el futuro. Gracias por ofrecerme este proyecto de mi interés y por toda la ayuda que me has ofrecido

Por último, un mensaje a todas las personas que puedan estar leyendo este documento. Gracias por leerme y espero que podáis encontrar en ello algo de ayuda.

Resumen

La administración de sistemas es un trabajo necesario en cualquier empresa. Todo proceso tecnológico, por básico que sea, conlleva un proceso de administración de sistemas, siendo este uno de los pilares fundamentales. Cualquier sistema informático o tecnológico debe estar actualizado, bien planificado, documentado y monitorizado para detectar y corregir posibles errores no deseados que puedan perjudicar a la entidad.

Lo que se pretende con este proyecto es dar una solución eficaz a una de las tareas básicas y más cotidianas a las que se enfrentan los administradores de sistemas, la instalación de nuevos equipos informáticos. No es ninguna creación nueva, pues el servicio PXE fue publicado por Intel en 1999; sin embargo, en este proyecto se ha mejorado su funcionalidad con el objetivo de que cualquier Sysadmin ahorre un tiempo muy valioso que puede ser utilizado para otras tareas.

La especificación del Preeboot eXecution Environment describe un sistema cliente-servidor estandarizado capaz de iniciar un conjunto de software, principalmente una imagen de un sistema operativo obtenida de una red. En este proyecto, se propone una herramienta gráfica multicast de clonado de sistemas por red. Con esta herramienta las piezas necesarias son equipos con una tarjeta de red compatible con PXE, un switch y un medio extraíble con el LiveCD de System Rescue CD en su versión 5.

En definitiva, se propone la implementación de una herramienta portable y aplicable a cualquier red de computadores en la que se desee clonar alguna partición de disco a otras máquinas destino. Bastará con hacer boot desde un medio extraíble en el equipo que deseamos que actúe como servidor y establecer en una sencilla interfaz gráfica las particiones de disco que deseamos enviar a equipos clientes. En la parte de los clientes estos son capaces de arrancar desde la red y sin interacción humana clonar a su disco duro interno las particiones elegidas en la parte del servidor.

Palabras Clave: Sysadmin, SystemRescueCD, PXE, multicast, instalación nuevos equipos.

Abstract

Systems administration is a necessary job in any company. Every technological process, however basic it may be, involves a systems administration process, which is one of the fundamental pillars. Any computer or technological system must be updated, well planned and documented and monitored to detect and correct possible unwanted errors that may harm the entity.

The aim of this project is to provide an effective solution to one of the basic and most common tasks faced by system administrators, the installation of new computer equipment. It is not a new creation, as the PXE service was published by Intel in 1999, however, in this project its functionality has been improved with the aim of saving any Sysadmin valuable time that can be used for other tasks.

The Preeboot eXecution Environment specification describes a standardized client-server system capable of booting a set of software, mainly an operating system image obtained from a network. In this Project i propose a graphical multicast tool for cloning systems over a network. With this tool, the necessary parts are computers with a PXE-compatible network card, a switch and a removable media with the SystemRescueCD LiveCD in its version 5.

In short, we propose the implementation of a portable tool applicable to any computer network where you want to clone a disk partition to other target machines. It will be enough to boot from a removable media in the computer that we want to act as server and establish in a simple graphical interface the disk partitions that we want to send to client computers. On the client side these are able to boot from the network and without human interaction clone to their internal hard disk the partitions chosen on the server side.

Keywords: Sysadmin, SystemRescueCD, PXE, multicast, installing new computers.

1. Introducción

Actualmente, las tareas de los administradores de sistemas crecen acorde a los avances tecnológicos. Las empresas y otros órganos deben acoger obligatoriamente estos nuevos cambios, ya que no se concibe ninguna industria o administración sin presencia de equipos informáticos [10] (Redigit, 2019). Cualquier tarea en un puesto de trabajo conllevará el uso de tecnología y todo lo que ello implica:

- Ordenadores de trabajo. En versión portátil, sobremesas, tablets, ultraligeros, convertibles, táctiles o PDAs. El presente proyecto se centra precisamente en la gestión y puesta en marcha de estos equipos personales o profesionales.
- Redes de conexión departamentales. Cableadas o inalámbricas. En el mundo del Internet en el que vivimos no tiene sentido disponer de ordenadores aislados. Entendemos por red, desde pequeñas redes para empresas o negocios usadas para el acceso a Internet, hasta grandes redes corporativas para el acceso a servicios de Intranet y de Internet. Los usos habituales de una red en la mayoría de las empresas son entre otras la comunicación externa e interna con proveedores externos o colaboradores. Por ejemplo, SAP, sistemas de ficheros en red, Microsoft Teams, telefonía IP interna o el uso de servicios cloud.
- Licenciamiento software y de servicios cloud.
- Paquetes software para el inventario, control y gestión de incidencias de todos estos dispositivos mediante herramientas como Service-Now o Jira.
- Técnicos de Infraestructura o administradores de sistemas. Esenciales para el funcionamiento de las partes anteriormente citadas. Las funciones básicas de estos expertos pasan por la puesta en marcha, monitorización, mantenimiento y resolución de incidencias y nuevas peticiones [5]. (Desde Linux, 2019)

El proyecto desarrollado busca dar respuesta a una de las tareas básicas de los administradores de sistemas, la puesta en marcha. Para ello, se implementará un sistema que permitirá a las empresas economizar recursos gracias al ahorro de tiempo.

1.1 Motivación

Como bien se ha destacado con anterioridad, la labor de los Administradores y técnicos de sistemas es cada vez más importante en el sector empresarial, independientemente de su tamaño, ya que la actividad desarrollada en estas por el personal o los equipos informáticos va a depender directamente de su buen funcionamiento. En la actualidad y como consecuencia de la globalización cualquier empresa, por pequeña que sea, utiliza cuentas de email o servicios de internet que implicarán una serie de equipos informáticos correctamente configurados por un administrador de sistemas.

La motivación técnica principal del proyecto no es otra que satisfacer una necesidad en la tecnología actual. En el mercado existen algunas soluciones que cubren parte de esta

necesidad, pero ninguna de ellas acaba de convencerme tanto como para liberar al técnico/administrador de sistemas de estas tareas que tanto tiempo requieren.

La motivación del desarrollo del proyecto es el desarrollo de una herramienta funcional que pueda ser usada en producción en cualquier entorno empresarial/administrativo. En mi corta experiencia profesional como técnico de sistemas he podido comprobar la gran cantidad de tiempo invertida en la instalación de equipos o en la puesta a punto o reparación de estos, por ejemplo, por la infección por malware o por discos duros dañados. Por ello, en este proyecto se plantea el desarrollo de una herramienta funcional y eficaz que permita ahorrar un valioso tiempo que se puede invertir en otras tareas.

1.2 Objetivo

El objetivo del proyecto es el desarrollo de una herramienta que permita el clonado de particiones y opciones de disco entre ordenadores pertenecientes a una misma red local.

Se mostrará el servicio PXE y todos sus componentes funcionando conjuntamente. Sin embargo, el proyecto no finaliza en este punto, sino que se busca ampliar los conocimientos en PXE, en el scripting Linux y en el Live CD de System Rescue CD. Como objetivo se busca conseguir un script de bash que desarrolle una herramienta gráfica para clonar particiones de disco de una máquina origen a un número x de máquinas destino clientes, todo ello con la mínima interacción humana posible.

Este trabajo pretende desarrollar una herramienta de clonado de ordenadores que permita:

- Aprovechar instalaciones limpias y bien configuradas de equipos.
- Que se pueda usar en toda una red local con tan sólo una unidad usb.
- Que sea sencilla y fácil de usar, incluso por alguien sin conocimientos avanzados.
- Que sea interactiva y llamativa al usuario.
- Que el tiempo requerido para su uso sea el menor posible.
- Que aproveche al máximo el rendimiento de la red.

El objetivo es la elaboración de un script que se ejecutará sobre el Live CD de System Rescue CD para automatizar la tarea de clonación de un único sistema a un número x de clientes. Con la idea propuesta en este proyecto se invierte el tiempo una vez en la creación de una herramienta que después se podrá usar en cualquier tipo de máquina o arquitectura que se tenga que replicar con una mínima inversión de tiempo.

2. Herramientas y servicios utilizados

En este apartado se enumeran y se describen de una manera simplificada las herramientas y los servicios utilizados para el desarrollo del proyecto. En el caso de los servicios, éstos pueden funcionar de manera independiente, pero es necesario su funcionamiento conjunto para lograr el objetivo final del proyecto.

2.1 Herramientas

En primer lugar, se describen las herramientas software utilizadas. Se trata de un gestor de máquinas virtuales (VirtualBox), de la tecnología de los Live-CDs y de System Rescue CD en concreto, de la programación software basada en un script de Bash, del paquete “Dialog” para el desarrollo de la interfaz gráfica de usuario (GUI) y de la herramienta UDP-Sender para la sincronización y transferencia de datos entre las máquinas.

2.1.1 VirtualBox

Oracle VM VirtualBox (conocido comúnmente como VirtualBox) es un software de virtualización para arquitecturas x86/amd64. Actualmente en desarrollo por Oracle Corporation como parte de su familia de software de virtualización. Por medio de esta aplicación corriendo sobre un sistema operativo anfitrión es posible de una manera simple crear máquinas virtuales como si de una física se tratase. Estas máquinas virtuales cogen del sistema operativo anfitrión los recursos hardware solicitados (CPU, memoria RAM o espacio de disco para crear una unidad virtual). Como sistemas operativos anfitriones compatibles tenemos GNU/Linux, Mac OS X, Windows y Solaris. Dentro de estos anfitriones podremos virtualizar prácticamente todos los sistemas operativos disponibles compatibles con la arquitectura x86/amd64.

2.1.2 Live-CDs

Un Live CD, también llamado live-distro es un sistema operativo autónomo almacenado y focalizado a un medio extraíble. Se le puede llamar autónomo al contener en el medio extraíble todos los datos necesarios para arrancar un sistema operativo completo y funcional sin la necesidad de instalar o tocar nada del disco. Aun así, es habitual en la mayoría de los Live CDs que incluyan las herramientas necesarias para poder instalarlos en el disco duro y poder testear el rendimiento real de la distribución elegida ya que siempre va a ser mayor la velocidad de transferencia de un disco duro del sistema que la de un medio extraíble.

La gran ventaja de los Live-CDs radica en su portabilidad entre diferentes máquinas sin tener que preocuparse por los detalles hardware específicos de cada máquina. Por otra parte, permite arrancar incluso en máquinas sin disco duro o cuando éste se encuentre dañado. Entre los usos más comunes de este tipo de distribuciones encontramos:

- Disco de demostración. Por ejemplo, para probar una distribución Linux y navegar por sus menús y herramientas antes de lanzarnos a instalarlo en nuestro disco duro.
- Disco de rescate. Para, en caso de tener problemas, arrancar un sistema operativo completo y acceder al disco duro para tratar de repararlo o recuperar ficheros importantes.
- Limpieza de amenazas. En caso de infección podemos arrancar un Live-CD que consideramos que no se encuentra contaminado y tratar de eliminar de manera manual la infección.
- Navegar de manera anónima. Ya que no es necesario instalar ningún tipo de archivo tampoco se guarda ningún registro. Una vez apagada la máquina toda actividad desaparecerá.

De entre las múltiples opciones de Live-CDs existentes en el mercado la elegida para este proyecto ha sido System Rescue CD en su versión v5.3 dada su gran compatibilidad con la herramienta desarrollada en este proyecto y que se describirá en esta memoria [13] (System Rescue, s. f.). También ha sido elegida esta distribución al ser una de las que cuenta con mayor documentación y comunidad online.

2.1.3 Programación en shell

En informática, el shell o intérprete de órdenes es el programa informático que provee una interfaz de usuario para acceder a los servicios del sistema operativo. El intérprete de órdenes generalmente se ejecuta en un dispositivo terminal con soporte para texto donde el usuario escribe órdenes; sin embargo, también puede leer y ejecutar órdenes desde un archivo que comúnmente llamamos “script”. Para el proyecto se ha generado un fichero (script) que contiene una serie de órdenes donde se han utilizado tuberías, sustitución de comandos, variables, pruebas de condición e iteración o programación basada en funciones.

Para el desarrollo del proyecto se ha escogido la Shell Bash. GNU Bash (Bourne-again shell) es una interfaz de usuario de línea de comandos lanzada por primera vez en 1989 y ampliamente utilizada como el intérprete de inicio de sesión (login) predeterminado en la mayoría de las distribuciones GNU/Linux [1] (About: Bash, s. f.). Se ha elegido la Shell de Bash debido a que se trata del intérprete con el que más familiarizado me encuentro y que cuenta con una gran comunidad y documentación online donde poder apoyarme.

2.1.4 Dialog

Se trata de una herramienta disponible para GNU/Linux y que se encuentra incluida en la distribución de System Rescue CD. Para el resto de las distribuciones se puede descargar e instalar desde los repositorios de la distribución utilizada.

Dialog es un comando de GNU/Linux que permite crear cuadros de diálogo en la terminal para utilizarse en los scripts de programación y mostrar o solicitar información al usuario de una forma gráfica. Este comando es usado en el proyecto para mostrar al usuario los discos duros del sistema y brindarle la opción de elegir un disco entre los disponibles en el sistema para después mostrarle las particiones de este y que pueda seleccionar las que desea enviar a las máquinas clientes.

2.1.5 UDP-Sender

Udp-sender es una herramienta disponible en distribuciones GNU/Linux usada para el multi-cast de ficheros a múltiples “udp-receivers” pertenecientes a la misma red LAN. La herramienta UDP-Sender usa Ethernet multi-cast. Esto conlleva que la eficiencia de red es la máxima posible al poder enviar información simultáneamente a muchas máquinas a la máxima velocidad posible sin depender esta del número de clientes recibiendo información.

2.2 Servicios

Entendemos por servicios aquellos componentes que nos ofrece el sistema operativo para resolver una necesidad. En el caso concreto de este proyecto necesitaremos 3 servicios que nos ofrece System Rescue CD.

2.2.1 Dhcpd

Dynamic configuration service protocol (DHCP), definido en los RFCs 3131, 2132 y 3315 para soporte de IPv6. Se trata de un protocolo de red usado ampliamente para la configuración dinámica de los parámetros de red de los hosts pertenecientes a una red determinada. Inicialmente desarrollado en 1985 como extensión del protocolo Bootstrap para conectar estaciones de trabajo sin disco duro con un Bootserver del que reciben su sistema operativo [7] (Ionos, 2019).

La asignación de direcciones IP y/o otros parámetros de red usando DHCP se basa en un modelo cliente-servidor, donde el host que quiere obtener su configuración realiza una petición broadcast. Dicha petición es recibida por un servidor DHCP central que a su vez recurre internamente a una base de datos que contiene los parámetros de red a asignar. Este pequeño servicio normalmente corre sobre un componente hardware de red o en red, como son los Routers o encaminadores. Entre los parámetros de red que este servicio puede proporcionar a un cliente tenemos:

- Dirección IP única (con un tiempo de lease).
- Máscara de subred.
- Puerta de enlace de red.
- Servidores DNS.
- NTP servers.
- WINS, proxy and X Servers.
- TFTP y PXE network boot servers.

Para este proyecto es especialmente interesante el último apartado TFTP y PXE ya que DHCP es una extensión retrocompatible de Bootp, protocolo pensado inicialmente para el boot por red de hosts Unix. De esta forma DHCP permite enviar a los clientes una configuración de red completa incluyendo:

- Los parámetros de red ya indicados anteriormente.
- Pxeboot, que se describirá en posteriores apartados.
- Kernel + initrd (boot SO ramdisk), que también se detallará a continuación.

La secuencia que sigue la comunicación clientes – servidor DHCP según Ionos, 2019 es la siguiente:

1. Cliente DHCP envía un paquete DHCPDISCOVER a la dirección de broadcast 255.255.255.255 desde 0.0.0.0. De esta forma el cliente DHCP localiza el/los servidores DHCP en la red.
2. Todos los servidores DHCP que escuchan la petición en el puerto 67 responden la solicitud del cliente con un paquete de tipo DHCPOFFER que contiene los parámetros de red adecuados para el cliente.
3. El cliente DHCP escoge uno de los paquetes recibidos y responde al servidor correspondiente con un paquete DHCPREQUEST. De esta forma también quedan informados el resto de los servidores DHCP de que no han sido elegidos. En este momento el cliente ya ha confirmado sus parámetros de red que le fueron enviados anteriormente.
4. El servidor DHCP finaliza confirmando los parámetros TCP/IP y enviando el PXEBoot. Este paquete además contiene un acknowledged (DHCPACK) o un not acknowledged (DHCPNAK) en caso de que no contase con más direcciones IP para ofrecer.

Una vez configurado el cliente DHCP este ya tiene una configuración de red válida para poder formar parte de la red y poder usar sus servicios. Por su parte, el servidor DHCP guarda en su base de datos la configuración ofrecida al cliente junto a su dirección MAC, haciéndose de esta forma la configuración permanente.

2.2.2 Tftpd

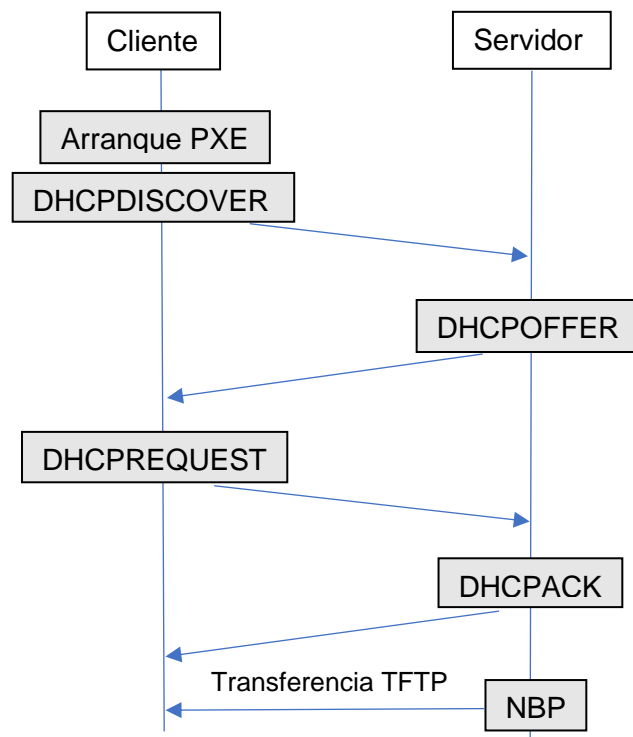


Figura 1. Diagrama de flujo simplificado PXE

Trivial File Transfer Protocol, llamado TFTP, es un protocolo cliente-servidor muy simple que regula la transferencia entre dos equipos informáticos. Originalmente definido en junio de 1981 en el RFC 783 y en su versión vigente publicada en 1992 en el estándar RFC 1350 o en el RFC 2347 de 1998. Por defecto TFTP usa el protocolo de transporte UDP (User Datagram Protocol) que nos ofrece la posibilidad de transmitir datos sin la necesidad de una conexión fija y/o control estricto de errores entre los miembros de la comunicación [6] (Ionos, 2019). Si bien TFTP sobre UDP nos ofrece una serie de ventajas para ser usado en PXE también se puede usar basándose en otros protocolos diferentes que nos brindarían otras opciones, como la seguridad en la conexión.

El protocolo TFTP funciona en base a usar 5 tipos de paquetes que empiezan con un campo de código de operación de 16 bits (Operations Code):

Código de operación	Tipo de paquete	Descripción
1	RRQ (Read request)	Solicitud de lectura
2	WRQ (Write request)	Solicitud de escritura
3	DATA (Data)	Paquete de datos
4	ACK (Acknowledgment)	Paquete de confirmación
5	ERROR (Error)	Paquete de error

Figura 2. Tipos de paquetes TFTP

La principal ventaja del uso de TFTP en el proyecto es su simplicidad, ya que permite la escritura y lectura de archivos sin la necesidad de establecer una conexión, siendo fácil de implementar y el precursor de la transferencia rápida de archivos. Sin embargo, también se debe destacar como limitación que se trata de un protocolo sin cifrado o mecanismos de autenticación, lo que hace asumir riesgos en el envío de archivos confidenciales, aunque esto no es un problema para el objetivo de este proyecto. De necesitarse la transferencia de archivos de una forma segura y confidencial disponemos de alternativas mucho más seguras como HTTPS.

En la [figura 1](#) podemos ver la secuencia DHCP-TFTP que llevan a cabo los clientes arrancando por red (PXE) en este proyecto:

1. Cliente DHCP lanza paquete DHCPDISCOVER extendido con las opciones de PXE al puerto UDP-67.
2. El servicio de redirección DHCP recibe el paquete DHCPDISCOVER extendido y responde con un paquete de difusión DHCPOFFER con las opciones PXE al puerto UDP-68. De esta forma la mayoría de los clientes DHCP-PXE auto configuran sus parámetros de red; estos clientes serán identificados con su GUID/UUID. Contenido de paquete DHCPOFFER extendido:
 - ➔ Campo PXE Discovery Control indicando si se debe usar Multicasting, Broadcasting o Unicasting para contactar con los servidores de arranque PXE.
 - ➔ Lista con las direcciones IP de los servidores de arranque PXE, ya que pueden coexistir varios en una misma red.
 - ➔ Un menú donde cada entrada representa un servidor de arranque PXE.

- ➔ Un prompt que indica por pantalla información útil al usuario. Por ejemplo: "Presione F8 para visualizar menú de arranque".
 - ➔ Un tiempo de espera máximo para lanzar la opción por defecto del menú.
3. El cliente DHCP-PXE elige el servidor de arranque PXE apropiado y el firmware envía un paquete DHCPREQUEST extendido de tipo multicast o unicast al puerto UDP-4011 o al puerto UDP-67 en caso de estar usando Broadcast. Este paquete contiene:
 - ➔ Servidor de arranque PXE.
 - ➔ Capa de arranque PXE que permitirá ejecutar múltiples tipos de servidores de arranque mediante un único programa de arranque.
 4. El servidor de arranque PXE recibe el paquete DHCPREQUEST extendido descrito y devuelve un paquete DHCPACK extendido con opciones PXE específicas:
 - ➔ Ruta completa del programa de arranque (Network Bootstrap Program, NBP) para posteriormente descargarle usando el protocolo TFTP.
 - ➔ Tipo de servidor de arranque PXE y la capa de arranque PXE.
 - ➔ Configuración multicast TFTP si se utilizase.

En este apartado es importante destacar que un servidor de arranque PXE debe soportar BIS (Boot Integrity Services), lo que permite al cliente PXE verificar el NBP descargado mediante un archivo checksum desde el mismo servidor.
 5. Tras recibir el paquete DHCPACK el NBP es descargado mediante TFTP y es ejecutado en la RAM del cliente para cargar o instalar el sistema operativo. Los archivos necesarios que serán transferidos serán descritos en posteriores apartados.

A continuación, se detallará el servicio PXE, es importante destacar que PXE no existe como servicio independiente, sino que es la combinación de los protocolos DHCP y TFTP.

PXE, de Preboot eXecution Environment es un entorno de ejecución de prearranque para computadoras, es decir, es un entorno de ejecución antes del arranque desde el disco para la propia tarjeta de red. PXE permite a la tarjeta de red iniciar un software propio (firmware) sin la existencia de un Sistema Operativo cargado en RAM y que permite consultar la existencia de servidores DHCP en red que puedan proporcionar información de ayuda para arrancar el equipo. No debemos confundir este entorno con WOL (Wake On Lan) usado para otros propósitos.

PXE fue introducido como parte del framework Wired for Management por Intel y fue descrito en la especificación (versión 2.1) publicada por Intel y Systemsoft en septiembre de 1999. PXE a su vez hace uso de manera de otros protocolos de red como pueden ser IP, UDP, DHCP o TFTP y de algunos conceptos usados en otros campos como GUID (Globally Unique Identifier), UUID (Universally Unique Identifier) o UNDI (Universal Network Device Interface). Importante destacar que el entorno PXE no sufre cambios o modificaciones desde su versión inicial de 1999. Desde un principio fue diseñado para funcionar sobre diferentes arquitecturas incluyendo las más comunes DEC Alpha o Intel IA-32. Un poco más recientemente junto al lanzamiento de UEFI (Unified Extensible Firmware Interface, en sustitución de las tradicionales BIOS) en 2002 Intel incluyó PXE

en la UEFI para la arquitectura IA-64 que predomina hasta la actualidad, creando un estándar de facto con esta implementación.

El protocolo PXE consiste en una combinación de los protocolos DHCP y TFTP. DHCP en la parte cliente se encarga de localizar el servidor de arranque apropiado, por lo que este debe estar correctamente configurado para ofrecer esta información a los clientes. Por otra parte, TFTP se encarga de la descarga del programa inicial de Bootstrap (secuencia o programa de arranque) y algunos archivos adicionales desde el servidor de arranque indicado por el DHCP previamente. Para iniciar una sesión de arranque con PXE el firmware del cliente envía un paquete de tipo DHCPDISCOVER con algunas opciones específicas de PXE al puerto UDP 67 por radiofusión. De esta forma, el cliente está indicando al servidor DHCP que su firmware es capaz y quiere arrancar usando PXE, siendo ignorados por servidores DHCP estándar, ya que no debemos olvidar que en una misma red pueden existir varios servidores DHCP con diferentes propósitos [8] (Linkfang, 2021)

2.2.3 Syslinux

El proyecto Syslinux está formado por un conjunto de gestores de arranque ligeros para arrancar un sistema operativo Linux.

El sistema empleado en el proyecto es Pxelinux. Es utilizado para arrancar un sistema Linux (System Rescue CD) desde un servidor de red usando el sistema PXE (entorno de ejecución de Pre-arranque). Pxelinux se define como un NBP (Network Bootstrap Protocol) altamente configurable que permite mostrar al usuario una línea de comandos o menús para arrancar un sistema operativo completo.

2.2.4 Secuencia de arranque Linux

A continuación, se describe el proceso de arranque de un sistema operativo Linux desde que se pulsa el botón de encendido [9] (Proceso de arranque de Linux, 2020):

1. BIOS. Al arrancar, el BIOS leerá la información MBR (Master Boot Record) del primer sector del disco duro.
2. Leer MBR. Uno de los datos almacenados por el MBR es el cargador de arranque (BootLoader). El propósito del inicio del BIOS es iniciar el cargador de arranque en el MBR.
3. Cargador de arranque. El cargador de arranque más utilizado en la actualidad es Grub, se trata de un software que lee los archivos del kernel y hace que se ejecuten, transfiriendo el control a éste.
4. Carga del kernel. El kernel es la parte más importante de un sistema Linux, su función es detectar el hardware y cargar los controladores más básicos.
5. Carga del initrd. Ya que el kernel se almacena en el disco duro en forma de archivos comprimidos, éste solo contiene los módulos más básicos. El resto de los controladores se cargan desde el archivo initrd. Este archivo es generado cuando se instala el sistema operativo y es un sistema de archivos raíz temporal

- (rootfs). Una vez se han cargado todos los controladores, el kernel llama activamente al proceso de inicio.
6. Inicio proceso de inicialización. Init es el primer proceso iniciado por el sistema, éste prepara el entorno del sistema operativo para la ejecución del software. Una vez el entorno operativo del sistema está listo se inicia el servicio del sistema.
 7. Inicio de los servicios. Se ejecutan los directorios correspondientes según el nivel definido (runlevel).
 8. Programa de inicio definido por el usuario. Tras completar el inicio de los diferentes servicios el sistema finalmente ejecuta el script `/etc/rc.d/rc.local`. Llegados a este punto, el inicio del sistema operativo está completo y se puede ver la interfaz de inicio de sesión.

En la [figura 3](#) podemos visualizar de una manera resumida los pasos del proceso de arranque por PXE de System Rescue CD.

3. Arranque por red de System Rescue CD

En este apartado se describirán las configuraciones y ficheros necesarios para que máquinas clientes puedan hacer arranque por red usando los servicios mencionados en el capítulo anterior, en esencia Dhcpd y Tftpd. He de destacar que las configuraciones descritas para arrancar System Rescue CD por red son válidas para arrancar otras distribuciones cambiando algunas opciones de los ficheros que se describirán y cambiando las imágenes comprimidas de la distribución. De igual forma, es posible dar la opción de elegir entre varias distribuciones la que se desea arrancar.

Como se describe gráficamente en la [figura 3](#) y como se ha explicado brevemente en el [capítulo 2.2.4](#) la secuencia que siguen los potenciales clientes del arranque por red es:

1. El lanzador (normalmente el firmware de la tarjeta de red) obtiene por DHCP los parámetros de red y el nombre del NBP. El escogido para este proyecto es pxelinux.0.
2. El gestor de arranque descarga los ficheros de configuración para poder mostrar al usuario las opciones de arranque. Estos ficheros se encuentran en el directorio “/srv/tftp/pxelinux.cfg” del equipo servidor.
3. El usuario elige la opción que quiere arrancar mediante el menú o el Prompt (estas opciones las contiene el fichero de configuración “Default”).
4. Una vez elegida la opción que se desea arrancar se descarga su kernel e initrd para después descargar y lanzar su sistema de ficheros comprimido.

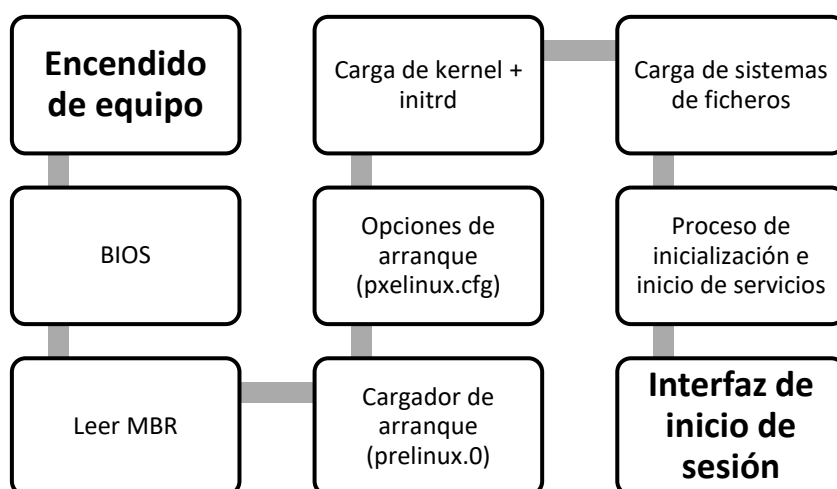


Figura 3. Secuencia arranque por red

En este siguiente apartado se describe en detalle los ficheros de configuración necesarios para que equipos clientes puedan arrancar por red un sistema operativo que necesita un kernel y una imagen de disco inicial (initrd).

3.1 Ficheros necesarios

Como se ha comentado en apartados anteriores, el servicio DHCPD indicará al cliente PXE el nombre del gestor de arranque y el servidor que contiene los ficheros necesarios

para arrancar la distribución que se elija. Para la transferencia de estos ficheros se usará el servicio TFTP, se trata de los siguientes ficheros que por defecto deben ubicarse en la ruta “/srv/tftp/” [12] (Syslinux, 2020):

- **Pxlinux.0** → Es un gestor de arranque (perteneciente al proyecto syslinux) ampliamente usado y que indica al sistema a que fichero debe dirigirse para obtener las configuraciones necesarias y el kernel del sistema operativo que se va a arrancar.
- **/pxlinux.cfg/** → Este directorio contiene los archivos de configuración para crear la interfaz gráfica (GUI) o la consola visualizada en los equipos cliente una vez cargado el gestor de arranque y transferidos estos ficheros de configuración.

Para crear una interfaz gráfica se deben crear y reemplazar en los lugares adecuados unos ficheros con extensión .c32 que contienen el estilo del entorno gráfico. El fichero por defecto que se cargará siempre, si no existen otros, debe llamarse “*Default*”. Cada entrada (Label) de este fichero contiene las rutas a los archivos necesarios para arrancar el sistema operativo y algunas opciones de cómo hacerlo:

- Label < > → Son las palabras mostradas al usuario en la interfaz para que identifique la opción que va a elegir.
- Kernel → Núcleo del sistema operativo.
- Append → Opciones y rutas con ficheros de configuración que se pasan al núcleo. Estas opciones son diferentes para cada tipo de Sistema operativo y deberemos ver las específicas de cada distribución en su documentación
- Initrd → Imagen del sistema operativo.
- Sistema de ficheros comprimido → Sistema de ficheros completo del sistema operativo.
- Autorun → Contiene la ruta del script que ejecutará el servicio Autorun justo antes de la carga de la consola de login.

En la [figura 4](#) podemos ver un ejemplo del fichero Default usado en este proyecto.

El gestor de arranque pxlinux nos permite mostrar al usuario 3 tipos de presentaciones para elegir el sistema operativo deseado:

- Línea de prompt básica → Muestra las opciones en un sencillo menú gráfico. Es la utilizada para el proyecto.
 - Menú en modo texto → Para elegir el Sistema Operativo utilizando los cursores a través de un menú en modo texto.
 - Menú en modo texto-gráfico → igual que la opción anterior, pero añadiendo una imagen de fondo.
- **/directorio_SO_x/** → Directorio donde almacenar la imagen completa de sistema operativo. Se deben organizar las distribuciones de esta forma para mantener ordenados los ficheros.

Por su parte, los clientes que arrancarán por red buscarán dentro del directorio “/pxelinux.cfg/” ficheros en el siguiente orden:

1. **pxelinux.cfg/xx-xx-xx-xx-xx-xx** → Es decir, una MAC de tarjeta de red.
2. **pxelinux.cfg/C0A80001** → IP de un cliente en hexadecimal.
3. **Pxelinux.cfg/Default** → Fichero que se carga por defecto en ausencia de los anteriores.

Esta configuración permite en función del cliente (identificado por su MAC de la tarjeta de red o su IP) cargar un fichero u otro. Esto por ejemplo nos permite configurar en el servicio dhcpd unas direcciones IP específicas a determinados clientes para después con estos ficheros crear una especie de lista blanca o negra de arranque por red.

La secuencia de carga habitual de cualquier sistema operativo consiste en:

1. Se carga el núcleo del sistema operativo que contiene lo más básico.
2. Se carga un disco RAM inicial (imagen initrd) que contiene un mini sistema de ficheros que contiene módulos que no se encuentran dentro del núcleo y drivers adicionales como el de la tarjeta de red.
3. Se carga el Kernel que contiene el sistema de ficheros completo del sistema operativo.

```
....  
LABEL rescuecd_std  
MENU LABEL 1) SystemRescueCd: default boot options  
KERNEL ifcpu64.c32  
APPEND rescue64 scandelay=5 netboot=http://192.168.1.5/sysrcd.dat setkmap=es  
ar_source=http://192.168.1.5/ping_uc/ -- rescue32 scandelay=5  
netboot=http://192.168.1.5/sysrcd.dat setkmap=es  
...
```

Figura 4. Fichero "Default" arranque por red SystemRescueCD

3.2 Servicio PXE/TFTP en SRCD versión 5

El primer paso imprescindible para cualquier servicio que haga uso de la red (dhcpd y tftpd) es configurar los parámetros de red de la máquina. System Rescue CD como la mayoría de las distribuciones Linux nos permite realizar esto de dos formas [10] (System Rescue, s. f.):

1. Mediante un gestor de red gráfico (uno de ellos):
Por ejemplo, podemos ejecutarlo con el comando: *\$net-setup*

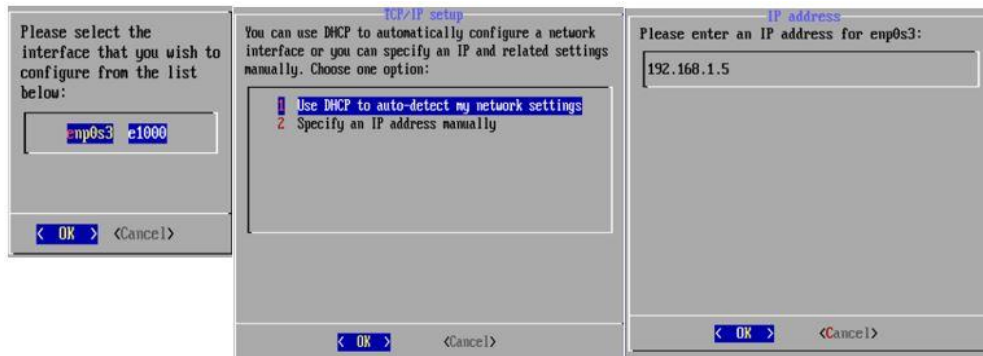


Figura 5. Gestor de red tráfico SystemRescueCD

2. Configuración desde línea de comandos:

`$ifconfig enp0s3 192.168.1.5` → `enp0s3` es variable y dependiente del nombre otorgado a la tarjeta de red.

`$route add default gw 192.168.1.1` → Configura el Gateway.

Una vez configurados los parámetros de red debemos emplazar los ficheros comentados en anteriores capítulos en sus respectivas ubicaciones:

- Fichero “**Default**”, en la ruta `/tftpboot/pxelinux.cfg`
- Script “**autorun0**”, en la ruta `/tftpboot/ping_uc`

Una vez configurada la red y con los ficheros de configuración en sus respectivas ubicaciones ya podemos arrancar el servicio PXE que nos brinda System Rescue CD:

`$/etc/init.d/pxebootsrv start`

```
root@sysresccd /tftpboot % /etc/init.d/pxebootsrv start
* Caching service dependencies ... [ ok ]
* Starting the pxe-boot-server ...
* /var/lib/dhcp: correcting owner
* /var/lib/dhcp/dhcpd.leases: creating file
* /var/lib/dhcp/dhcpd.leases: correcting owner
* Starting dhcpd ... [ ok ]
* Starting tftpd ... [ ok ]
* Starting tftpd ... [ ok ]
```

Figura 6. Arranque servicio PXE SystemRescueCD

Tal y como se observa en la [figura 6](#) y como se ha comentado en anteriores capítulos el servicio pxe hace uso de los servicios dhcpd, httpd y tftpd. La versión 5 de System Rescue CD proporciona la herramienta (`/etc/init.d/pxebootsrv`) que automatiza el despliegue de estos servicios. Esta herramienta libera de configurar manualmente los servicios involucrados para el correcto funcionamiento del pxe y que se deberían configurar de forma manual si usásemos otras distribuciones [3] (Debian Wiki, 2019).

4. Ingeniería software del proyecto

Entendemos ingeniería del software como la disciplina formada por un conjunto de métodos, herramientas y técnicas que se utilizan en el desarrollo de cualquier aplicación informática con el objetivo de desarrollar un software de calidad, bien planificado y que resuelva el problema planteado.

En este capítulo se describirá la metodología de trabajo empleada y se nombrarán los requisitos identificados al inicio del proyecto.

4.1 Metodología de trabajo usada

Una metodología de desarrollo software se define como una serie de fases por las cuales debería pasar cualquier proyecto software, con el objetivo principal de dar al proceso una estructuración y planificación. Existen numerosas metodologías de trabajo adaptándose cada una de ellas al tipo de proyecto que se esté desarrollando. Para la realización de este proyecto se ha utilizado el modelo de desarrollo incremental, al tratarse del modelo que mejor se adapta al proyecto y a mi estilo de programación. En el desarrollo del proyecto, ha sido posible dividir este en partes. Cada una de estas partes se han podido desarrollar y probar independientemente para ir integrándolas al proyecto hasta satisfacer los requisitos funcionales identificados.

El modelo de desarrollo incremental, también llamado iterativo, consiste en la división del proyecto en partes independientes que se van integrando al proyecto aportando un crecimiento progresivo de la funcionalidad de la aplicación.

Las principales ventajas aportadas por este modelo al proyecto son:

- Generar software operativo desde un principio. Permitiendo probar cada una de las partes independientemente.
- Separar la complejidad del proyecto al dividirlo en partes más pequeñas y manejables. Esto aporta la capacidad de poder centrarnos en cada parte, aportando un mayor aprendizaje de cada iteración.
- Mayor facilidad para gestionar posibles cambios o ampliaciones que puedan surgir al no tener que modificar los componentes ya desarrollados.
- Posibilidad de reutilización del software para otros proyectos al disponer de pequeños componentes funcionales que resuelven un problema concreto.

4.2 Requisitos funcionales

Los requisitos funcionales son aquellos que describen las funcionalidades de la aplicación entendidos como las actividades que los usuarios van a poder realizar usando la herramienta desarrollada. Los requisitos funcionales identificados son representados en la [figura 7](#) con su correspondiente identificador y descripción.

Identificador	Descripción del requisito
RF-01-USB	El usuario podrá iniciar el sistema de clonado arrancando el ordenador origen con un medio extraíble, por ejemplo, un USB.
RF-02-ListarDiscos	El usuario podrá consultar gráficamente información de los discos duros del sistema.
RF-03-ElegirDisco	El usuario podrá visualizar los discos duros del sistema y elegir uno de ellos.
RF-03-ElegirOpcionesFijas	El usuario podrá elegir al menos 3 partes del disco duro a clonar a las máquinas clientes (p: tabla de particiones, b: master boot record, g: grub)
RF-04-ElegirOpcionesDinámicas	El usuario podrá visualizar y elegir las particiones del disco duro elegido y que serán clonadas hacia los clientes.
RF-05-RevertirOpcionesElegidas	El usuario podrá visualizar las opciones elegidas y confirmarlas o volver a empezar su elección.

Figura 7. Requisitos funcionales

4.3 Requisitos no funcionales

Los requisitos no funcionales son aquellos que describen características de la aplicación no ligadas a su funcionalidad, sino aspectos de portabilidad, rendimiento, usabilidad... Los requisitos no funcionales identificados son representados en la siguiente tabla ([figura 8](#)) con su correspondiente identificador y descripción.

Identificador	Descripción del requisito
RNF-01-Amigable	La interfaz de la herramienta deberá ser llamativa y atractiva para el usuario.
RNF-02-Usable	La herramienta deberá ser intuitiva y usable incluso para alguien con conocimientos básicos de informática.
RNF-02-Portable	La aplicación deberá ser portable a otras máquinas con arquitectura x86 o amd64.
RNF-03-GranRendimiento	La aplicación deberá funcionar para una red multicast para aprovechar al máximo posible los recursos de red.

Figura 8. Requisitos no funcionales

4.4 Requisitos hardware y software

En cualquier proyecto software es importante definir desde un principio cuales son los requisitos hardware y software que ha de satisfacer. De esta forma, se podrá desarrollar el proyecto desde un principio teniendo en cuenta dónde se va a utilizar el software desarrollado para así ir resolviendo cualquier dependencia. De otra forma, podríamos encontrarnos con software que funcionase bajo determinadas circunstancias pero que no funcionase en otras, por ejemplo, en sistemas operativos más modernos o en hardware de determinada arquitectura.

Para el desarrollo de este proyecto no es necesario describir ningún requisito software puesto que la herramienta desarrollada incluye todo componente software necesario, siendo así completamente portable a cualquier máquina que satisfaga los requisitos HW descritos a continuación ([figura 9](#)).

En cuanto a los requisitos hardware tenemos varios de estricto cumplimiento para el funcionamiento de la aplicación. Estos requisitos son descritos en la siguiente tabla (figura 9).

Identificador	Descripción del requisito a satisfacer.
RHW-01-Especificaciones	Arquitectura CPU x86 o x64, Pentium III o superior. 128 MB RAM.
RHW-02-TarjetaDeRed	Tarjeta de red (NIC) compatible con arranque por red (PXE) y compatible con System Rescue CD v5.3.
RHW-03-BIOS	BIOS compatible con arranque por red. Legacy boot y EFI.
RHW-04-TamañoDiscos	Discos duros de equipos clientes de igual o mayor tamaño que disco de equipo actuando de servidor.
RHW-05-MedioExtraible	Equipo/BIOS servidor compatible con arranque desde medio extraíble, CD-ROM o USB. CD-ROM 700 MB o memoria desde 1 GB de capacidad.
RHW-05-Red	Equipamiento básico de red, al menos un switch y latiguillos.

Figura 9. Requisitos hardware

4.5 Planificación

Para el desarrollo de este proyecto se ha dividido el trabajo en varias fases, siendo cada fase necesaria para la realización de la siguiente. Como se mencionó anteriormente en el capítulo 4.1, la metodología de trabajo ha sido incremental, de forma que cada nueva fase aquí descrita aporta una nueva funcionalidad a la herramienta desarrollada.

El desarrollo del proyecto, dividido por fases responde a la siguiente secuencia:

1. Investigación de herramientas comerciales para el clonado.
2. Investigación de gestores de arranque. Syslinux.
3. Investigación y puesta en marcha de Live-CD System Rescue CD.
4. Investigación y puesta en marcha de servicio PXE. Protocolos dhcp, tftp y ficheros de configuración.
5. Pruebas servicio DHCP, TFTP y PXE.
6. Desarrollo de script de clonado. Desarrollado por partes diferenciando el tipo de partición a clonar.
7. Pruebas de script de clonado modificando el script manualmente.
8. Investigación y desarrollo de interfaz gráfica de usuario (GUI) para la elección de opciones y particiones de disco.
9. Pruebas GUI.
10. Pruebas script de clonado usando la GUI desarrollada.
11. Pruebas herramienta usando diferentes escenarios de disco y varios clientes simultáneamente.

5. Desarrollo del script

En este capítulo se describe el desarrollo y las partes del script realizado que ejecutan los clientes PXE nada más terminar el arranque por red de System Rescue CD. El script será lanzado automáticamente en los clientes por el servicio Autorun de System Rescue CD, ya que así se ha indicado en el fichero [“Default”](#) en la opción “ar_source”, indicando la ruta exacta del fichero que contiene el código del script.

Autorun se trata de un servicio ofrecido por distribuciones Linux para la ejecución de scripts o comandos nada más finalizar la carga y arranque del sistema operativo. De esta forma, el comando configurado se ejecuta justo antes de la carga de la ventana de login del sistema. Este servicio resulta de gran utilidad para la automatización de tareas en el arranque.

Para el desarrollo del proyecto se han creado 3 máquinas virtuales interconectadas entre sí por una red NAT virtual creada sobre el propio VirtualBox. La primera de las máquinas virtualizadas es un equipo con 2 cores de CPU, 4 GB de RAM y un disco duro de 32 GB en el que se ha instalado un sistema operativo Windows 7 junto a un sistema Debian, por lo tanto, también se cuenta con un gestor de arranque (Grub). Las otras dos máquinas virtualizadas cuentan con 1 core asignado y 2 GB de memoria RAM. Cuentan con una unidad de disco de 32 GB y se encuentran conectadas a la misma red NAT virtual que la primera máquina que realizará el rol de servidor.

Como se comentó en el anterior capítulo, la metodología de desarrollo elegida ha sido la incremental, por esta razón, se ha desarrollado el script basándose en funciones, variables globales y un programa principal (main) que va invocando cada función en el orden adecuado. En este punto, cabe destacar que el mismo script (autorun0) es ejecutado por ambos roles, es decir, por la máquina con rol de servidor y por las máquinas clientes. En el caso del servidor el script es lanzado manualmente, y en caso de los clientes es lanzado automáticamente por el servicio Autorun. Por esta razón, como se puede ver en la siguiente figura, en el programa principal se diferencia entre si el script es lanzando manualmente en una máquina actuando de servidor o si por el contrario se está lanzando en un cliente (ejecutado por el servicio Autorun).

```
...
if grep ar_source /proc/cmdline; then #Client Side
    echo "Client Mode"
    echo "Cloning will start with options chosen in server side"
    scriptClone_SERVER_to_CLIENTS #Run clone script
else #Server side
    echo "Server Mode"
    infoLocalDisks_Temp=/tmp/infoLocalDisks_Temp
    trap "rm -f $infoLocalDisks_Temp" 0 1 2 5 15
    getInfoLocal_Disks
    graphics_GUI
fi
...
```

Figura 10. Programa principal (main)

5.1 Script de clonado

En este apartado se describe la función encargada de la clonación servidor-clientes de las opciones y particiones de disco elegidas. Esta función incluye dos tareas principalmente. Por un lado, la lógica de comunicación y sincronización de los clientes con el servidor ([capítulo 5.1.1](#)) y, por otro, la lógica para la transferencia de las opciones y particiones de disco elegidas hacia los clientes.

Al terminar esta función, esta puede ser utilizada independientemente sin la necesidad de otras funciones, esto se consigue escribiendo manualmente en el script las partes y opciones del disco que deseamos clonar hacia los clientes. Si no se realiza manualmente, estas variables de entrada son completadas por otras funciones que despliegan la interfaz gráfica de usuario que se describirá en el siguiente [capítulo 5.2](#). De esta forma, las variables de entrada (variables globales en el script) de la función de clonación son:

```
...
grub_part=$GRUB_PART
command=$master
dev=/dev/${CHOICE_D}
...
```

Figura 11. Opciones función clonar (GUI)

Sin la necesidad de usar las anteriores 3 variables globales (GRUB_PART, master(), CHOICE_D) rellenas con la interfaz gráfica de usuario, estas opciones se pueden rellenar manualmente. Por ejemplo, para hacer réplicas idénticas de una máquina con un único disco duro, 3 particiones de disco y dos sistemas operativos instalados (Windows + Debian) las opciones a elegir manualmente serían:

```
...
grub_part=3
command='master(p,b,g,1,2,3)
dev=/dev/sda
...
```

Figura 12. Opciones función clonar (ejemplo, manual)

Prestando atención a la figura anterior, en la opción “grub_part” se indica el número de partición que contiene un sistema de ficheros Linux desde el cual se cogerán los ficheros de instalación del gestor de arranque Grub, en el caso de que sea necesaria su instalación. En la siguiente línea “command=” se escribirá entre los paréntesis del comando master las partes y opciones del disco que se quieren transferir a los clientes. En el ejemplo expuesto, las opciones elegidas son todas las posibles (**p** → Tabla de partición del disco, **b** → MBR, bloque del master boot record, **g** → Grub, **1,2,3** → indican los números de particiones de disco). Por último, en la línea con la opción “dev=” se escribe el nombre del disco duro origen, en el ejemplo, como tan solo disponemos de un disco, este es “/dev/sda”.

Una vez el script dispone de las particiones y opciones de disco deseadas para clonar, el código evalúa si la máquina que se encuentra ejecutando el código es un cliente o es

un servidor. En consecuencia, genera otra secuencia de código que almacena en otro script para finalmente empezar su ejecución. Esta secuencia de código contiene las instrucciones para la comunicación y sincronización servidor-clientes y las instrucciones para clonar las particiones y opciones de disco elegidas con anterioridad (bien sea manualmente o usando la GUI que se mostrará en el [capítulo 5.2](#)).

```
# Location of the master script. This is the one in charge of sending in the correct order
# the file systems of the selected partitions.
master_script=/tmp/master_script_1.sh

# Location of the slave script. This is generated by the master and transmitted to all
# slaves to be executed.
slave_script=/tmp/slave_script_1.sh
... # Code that generate master and slave script
echo executing slave script...
. $slave_script
echo finished slave script...

...
echo executing master script...
. $master_script
echo finished master script...

...
```

Figura 13. Generación y ejecución script master y esclavo

5.1.1 Comunicación servidor-clientes (UDP-Sender)

Las comunicaciones entre el servidor y los clientes se realizan usando el servicio UDP-Sender descrito en el [capítulo 2.1.5](#). Tanto el script “master”, como el script “slave” comentados en el anterior capítulo, contienen las instrucciones para abrir comunicación entre dos puntos y realizar la transferencia de la información correspondiente.

Para el envío de información, se usa el comando “udp-sender fichero”, donde “fichero” es la información que se envía. De esta forma, todos los clientes que se encuentren escuchando por “udp-receiver” abren la comunicación y reciben la información, pues recordemos que es una comunicación multicast. Usando este esquema de udp-sender/udp-receiver se transfieren todas las particiones y opciones de disco elegidas. Por cada instrucción “udp-sender” en el script master, debe existir otra instrucción análoga en el script esclavo. Veamos un ejemplo en la siguiente figura:

```
...
cat <<-EOF >> $master_script
echo sending ntfs filesystem from $dev$part
ntfscclone -f -s -o - $dev$part | pv -W -s $size -cN $dev$part | \
udp-sender $udp_sender_args
EOF
...
cat <<-EOF >> $slave_script
echo receiving ntfs filesystem for $dev$part
udp-receiver $udp_receiver_args | pv -W -s $size -cN $dev$part | \
ntfscclone -r -O $dev$part - 2>&1
EOF
```

...

Figura 14. Comunicación cliente-servidor con UDP-Sender

Siguiendo este esquema, se transfieren una a una cada una de las opciones de disco y/o particiones de disco que se elijan. Por cada una de las opciones, se abre la conexión en la parte master con el comando “udp-sender” para después recibir la información en la parte del esclavo con el comando “udp-receiver”.

5.1.2 Clonado de particiones de disco seleccionadas

Una vez disponiendo del canal de comunicación entre servidor-clientes y sabiendo que información deseamos enviar, preparamos y enviamos los datos por el canal UDP-Sender abierto.

Como se ha mencionado anteriormente, los datos a enviar son completamente elegibles, pudiéndose enviar todas las opciones y particiones posibles (para un clonado de disco idéntico) o enviar opciones por separado a modo de recuperación. Por ejemplo, una sola partición, o reinstalar el gestor de arranque Grub para máquinas en las que conviven SOs Windows y Linux.

Dependiendo de las opciones elegidas, la lógica para preparar y enviar los datos a los clientes es diferente. Como se puede ver en la siguiente figura, en el desarrollo del script se diferencia entre las opciones elegidas con sentencias “if”. Dependiendo del tipo de dato/opción elegida (**tabla de particiones, mbr, grub y particiones de disco**) aplicamos una lógica u otra.

Por otra parte, dentro de las particiones de disco, también se diferencia entre tipos de particiones para aplicar lógicas diferentes. Se diferencia entre particiones **Type=83** → Correspondiente a una **partición Linux**, **Type=82** → **Partición Swap de Linux** o **Type=7** → Correspondiente a **partición NTFS**.

...

```
for part in $(echo ${command#master} | sed 's/[(),]/ /g'); do
    if [ $part == 'b' ]; then
        ...
    elif [ $part == 'g' ]; then
        ...
    elif [ $part == 'p' ]; then
        ...
    else
        type=$(sfdisk -qL --id $dev $part 2> /dev/null)
        if [ $type == 83 ]; then
            ...
            elif [ $type == 82 ]; then
                ...
            elif [ $type == 7 ]; then
                ...
            else
                echo ERROR: Do not know how to clone partition $part
            fi
        fi
    fi
done
```

...

Figura 15. Diferenciación opciones y tipos de particiones

Una vez diferenciadas las opciones y tipos de particiones, se generan la secuencia de comandos usada para preparar y enviar los datos. Los esquemas seguidos dependiendo de la opción y/o partición son:

- **part = b** → Se envía el único bloque (el primero del disco) del mbr usando el comando “**dd**” y el UDP-Sender.
- **part = g** → Los clientes reinstalan el gestor grub cogiendo los ficheros de instalación de la partición del máster que contiene un sistema Linux (esta partición es elegida por el usuario). Los clientes ejecutan *\$grub-install*.
- **part = p** → El máster hace un backup de la tabla de particiones usando *\$sfdisk -d* y se transfiere a los clientes para que restauren la copia en su disco.
- **type = 83** → El máster empaqueta la partición Linux usando el comando *\$tar -c ...* y se envía a los clientes para que hagan la operación inversa. En este apartado se usa la utilidad “PV (pipe viewer)” de forma que se muestre el progreso de la transferencia, dado que esta no es inmediata.
- **Type = 82** → El máster espera un tiempo prefijado para que los clientes creen y formateen la partición swap usando el comando *\$mkswap*.
- **Type = 7** → Se calcula el tamaño de la partición ntfs y después el máster hace un backup de la partición con *\$ntfsclone -f -s -o* y los clientes restauran el backup con *\$ntfsclone -r -O*. De nuevo, como en la parte de la partición Linux, se ha usado la utilidad “PV” para mostrar progreso de la transferencia.

Por último, se añaden en el script unas instrucciones para indicar al usuario que la partición elegida no es reconocida en el caso de que no sea un tipo de partición tratada en el script. De esta forma, excluimos aquellos tipos de particiones que no sean comúnmente empleadas en instalaciones de sistemas operativos. Por ejemplo, una partición de tipo FAT32 o las típicas particiones extendidas autogeneradas.

5.2 Interfaz gráfica de usuario (GUI)

Como se mencionó con anterioridad, a la función encargada de preparar y transferir los datos de disco deseados se le pueden indicar las opciones manualmente como se observa en la [Figura 12](#). Sin embargo, en este capítulo se desarrolla una interfaz gráfica de usuario (GUI). En esta, el usuario puede elegir las opciones y particiones de disco que desea clonar visual e interactivamente. A continuación, se mostrarán y describirán en el orden lógico de ejecución cada una de las ventanas gráficas desarrolladas en el proyecto:

1º. Nada más ejecutar la herramienta en una máquina actuando con el rol de servidor el usuario visualiza la [pantalla principal](#) de la interfaz gráfica de usuario. En esta ventana se muestran 3 opciones:

1. **Local disks Info** → Muestra información detallada de todos los discos duros presentes en el equipo.

2. **Prepare to handle remote clone requests** → Continúa la ejecución y permitirá al usuario elegir entre uno de los discos duros del sistema. Después podrá elegir las opciones y particiones a transferir a clientes, y podrá elegir la partición que contiene un sistema Linux desde donde instalar el gestor de arranque “grub”, en el caso de que se haya elegido su instalación.
3. **Exit** → Salir del script y volver a la consola de System Rescue CD ([Figura 16](#)).



Figura 16. Consola SRCD

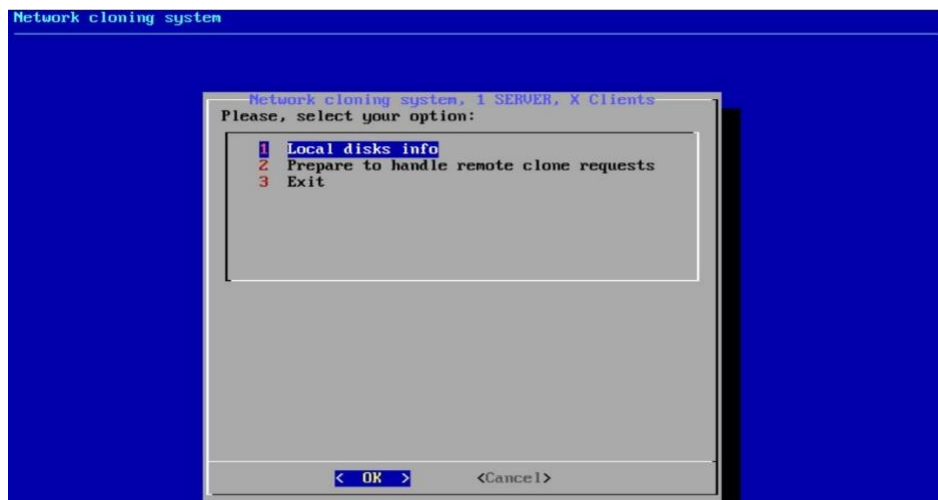


Figura 17. Pantalla principal GUI

2º. Como vemos en la siguiente figura, eligiendo la opción 1 (**Local disks Info**) se muestra información en detalle de los discos instalados. Presionamos “Exit” para volver a la pantalla principal.

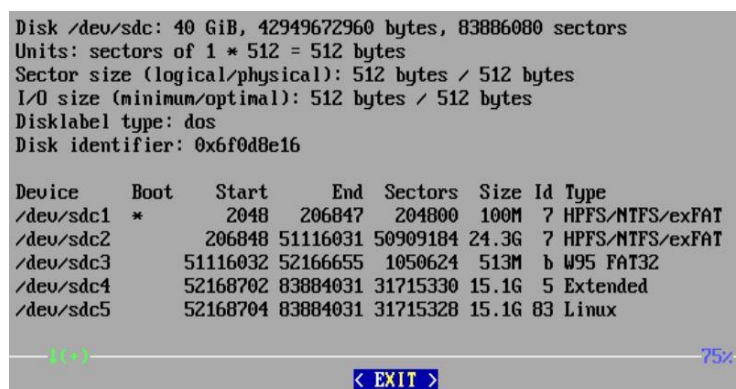


Figura 18. Información discos duros GUI

3º. En la siguiente figura vemos como eligiendo la opción 2 (**Prepare to handle remote clone requests**) se continúa el flujo del GUI y se permite al usuario elegir un disco duro objetivo entre todos los conectados y accesibles por el sistema.



Figura 19. Elegir disco duro GUI

4º. Una vez elegido el disco duro objetivo, en la siguiente ventana ([figura 20](#)) se muestran las opciones fijadas p -> Tabla de particiones, b -> mbr, g -> Grub y las particiones de disco que contiene el disco duro elegido en el anterior paso. El usuario debe seleccionar las opciones deseadas y presionar "OK" para continuar.

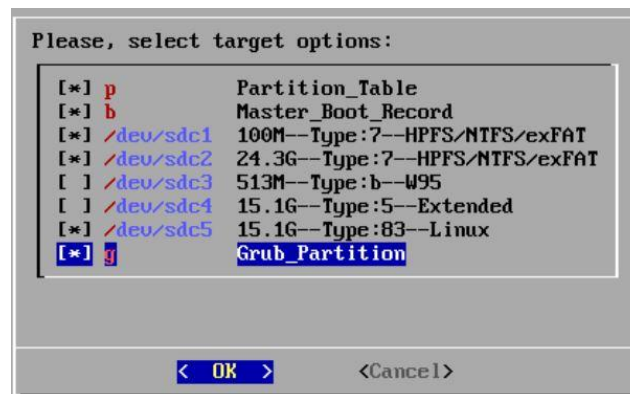


Figura 20. Elegir opciones GUI

5º. Si en el anterior paso se seleccionó la opción g (Grub), el usuario visualizará una nueva ventana ([figura 21](#)) donde deberá seleccionar desde qué partición leer los archivos de instalación del gestor de arranque Grub. Si anteriormente no se seleccionó "g", el usuario no visualizará esta ventana y pasará directamente a la ventana de confirmación ([6º](#)).

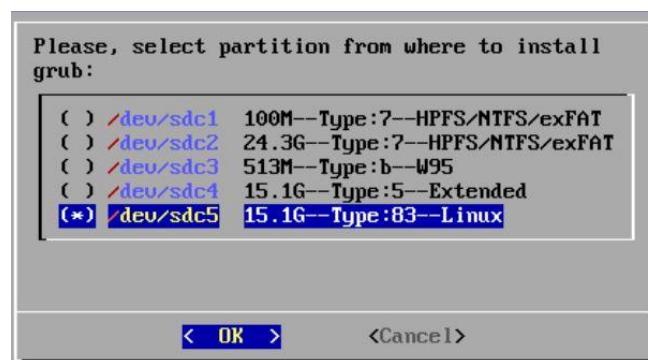


Figura 21. Elegir partición grub GUI

6º. Finalmente, el usuario visualiza en una nueva ventana las opciones elegidas y se le solicita confirmación ([figura 22](#)). Si se elige “yes”, se continua la ejecución del script con las opciones y particiones elegidas. Si por el contrario no está conforme o se ha equivocado en la selección, debe presionar “no” para volver a la [pantalla principal](#) sin continuar la ejecución del script.

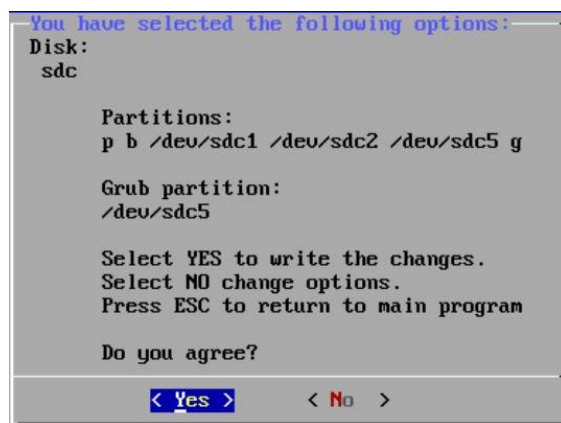


Figura 22. Ventana de confirmación GUI

En este apartado cabe mencionar que la interfaz gráfica de usuario (GUI) ha sido desarrollada usando el paquete “Dialog” incluido en System Rescue CD. “Dialog” permite crear ventanas para la visualización plana de información, como por ejemplo en la [opción 1](#) para la visualización del detalle de los discos duros del sistema. Permite crear ventanas con 1 (--radiolist) o varias (--checklist) opciones seleccionables, como, por ejemplo, para la [elección del disco duro objetivo](#).

A continuación, se expone a modo de ejemplo la función encargada de mostrar al usuario los discos duros del sistema para la elección de uno de ellos:

```
...
choose_Disk(){
    disks=$(lsblk -l --output TYPE,KNAME,SIZE | grep disk | sort | awk '{print
$2,$3,"off"}')
    CHOICE_D=$(dialog --radiolist "Please, select a target disk:" 0 0 0 $disks 3>&1
1>&2 2>&3)
}
...
```

Figura 23. Código elegir disco, Dialog - GUI

Como se puede observar en la figura, en la variable “disks” se lee y se almacena la información de los discos duros del sistema. La información almacenada es el nombre de la unidad (sda, sdb, sdc...), su tamaño, y la palabra especial “off”.

Por su parte, el comando “dialog” debe recibir la información en una lista con x elementos múltiplos de 3, el primero de ellos será el elemento guardado si es elegido, en este caso es almacenado en la variable global “CHOICE_D”. Los elementos 1 y 2 de

cada bloque de 3 datos serán los que el usuario visualizará en cada línea de la ventana creada. El tercer elemento predefinido “off” se usa para delimitar las líneas.

Siguiendo esta secuencia y tomando como ejemplo los discos de la [figura 20](#) el contenido de la variable “disks” sería:

```
disks= sda 31GB off sdb 12GB off sdc 40GB off sdd 42.4GB off
```

De esta forma, el comando “dialog” interpreta la palabra “off” como que debe hacer un salto de línea para seguir mostrando los siguientes elementos.

Siguiendo el mismo ejemplo de la [figura 19](#), si el usuario selecciona el tercer disco (sdc) el valor de la variable “CHOICE_D” sería:

```
CHOICE_D= sdc
```


6. Evaluación y pruebas

En este capítulo se mostrarán las pruebas realizadas para verificar el correcto funcionamiento de la herramienta desarrollada. Con las pruebas realizadas se ha comprobado la resolución del problema inicial planteado y que quedan satisfechos los requisitos nombrados en los [capítulos 4.2 y 4.3](#).

Es preciso comentar que, por simplicidad, las pruebas aquí mostradas se han desarrollado en un entorno virtual de VirtualBox con 3 equipos virtuales (1 servidor – 2 clientes) unidos entre sí por la misma red LAN. La misma herramienta ha sido probada en un entorno físico con 2 máquinas (cliente – servidor) resultando las pruebas satisfactorias al funcionar de igual forma. Las pruebas han sido realizadas con dos clientes, pero el funcionamiento y el rendimiento de la herramienta es el mismo con x clientes, ya que la herramienta ha sido pensada y desarrollada para ser multicast.

6.1 Prueba de servicio dhcpd

Como se comentó en la [figura 6](#) del [capítulo 3.2](#), System Rescue CD proporciona la herramienta necesaria para que con la ejecución de unos pocos comandos se tengan operativos los servicios DHCP y la combinación de varios protocolos que se entienden como el servicio PXE.

En la máquina con rol de servidor se comprueban que el servicio dhcpd se encuentra activo: `$etc/init.d/dhcpd status`

```
root@sysresccd /root % /etc/init.d/dhcpd status
* status: started
```

Figura 24. Estado servicio dhcpd

Se arrancan los dos clientes desde la tarjeta de red presionando la tecla “F12” en el arranque. Se comprueba que ambos clientes localizan al servidor DHCP y comienza el intercambio de mensajes descrito en el [capítulo 2.2.1](#). En las [figuras 25 y 26](#) se puede ver como los dos clientes arrancados por red obtienen una configuración de red desde el servidor DHCP (IPs: 192.168.1.100 y 192.168.1.101). Por otra parte, se observa cómo el DHCP también les proporciona el Pxeboot (pxelinux.0).

```
Cliente1 [Corriendo] - Oracle VM VirtualBox
iPXE 1.0.0+ -- Open Source Network Boot Firmware --
Features: DNS TFTP HTTP PXE PXEXT Menu

net0: 08:00:27:0a:f8:16 using 82540em on PCI00:03.0
[Link:down, TX:0 TXE:0 RX:0 RXE:0]
[Link status: Down (http://ipxe.org/38086101)]
Waiting for link-up on net0... ok
DHCP (net0 08:00:27:0a:f8:16)..... ok
net0: 192.168.1.100/255.255.255.0 gw 192.168.1.254
Next server: 192.168.1.5
Filename: /pxelinux.0
tftp://192.168.1.5/pxelinux.0... ok
```

Figura 25. Arranque DHCP cliente1

```
Cliente2 [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
iPXE 1.0.0+ -- Open Source Network Boot Firmware --
Features: DNS TFTP HTTP PXE PXEXT Menu

net0: 08:00:27:d1:26:4b using 82540em on PCI00:03.0
[Link:down, TX:0 TXE:0 RX:0 RXE:0]
[Link status: Down (http://ipxe.org/38086101)]
Waiting for link-up on net0... ok
DHCP (net0 08:00:27:d1:26:4b)..... ok
net0: 192.168.1.101/255.255.255.0 gw 192.168.1.254
Next server: 192.168.1.5
Filename: /pxelinux.0
tftp://192.168.1.5/pxelinux.0... ok
```

Figura 26. Arranque DHCP cliente2

En la parte del servidor, se visualiza en la [figura 27](#) el log del servicio DHCP. En este log comprobamos que recibe y envía a los clientes DHCP los paquetes correspondientes (DHCPDISCOVER, DHCPOFFER, DHCPREQUEST y DHCPACK). Para acceder al log usamos el comando: `$tail -f /var/log/messages`

```
Jun 28 12:43:17 sysresccd dhcpd[3047]: DHCPDISCOVER from 08:00:27:0a:f8:16 via enp0s3
Jun 28 12:43:18 sysresccd dhcpd[3047]: DHCPOFFER on 192.168.1.100 to 08:00:27:0a:f8:16 via enp0s3
Jun 28 12:43:20 sysresccd dhcpd[3047]: DHCPREQUEST for 192.168.1.100 (192.168.1.5) from 08:00:27:0a:f8:16 via enp0s3
Jun 28 12:43:20 sysresccd dhcpd[3047]: DHCPACK on 192.168.1.100 to 08:00:27:0a:f8:16 via enp0s3
Jun 28 12:43:37 sysresccd dhcpd[3047]: DHCPDISCOVER from 08:00:27:d1:26:4b via enp0s3
Jun 28 12:43:38 sysresccd dhcpd[3047]: DHCPOFFER on 192.168.1.101 to 08:00:27:d1:26:4b via enp0s3
Jun 28 12:43:40 sysresccd dhcpd[3047]: DHCPREQUEST for 192.168.1.101 (192.168.1.5) from 08:00:27:d1:26:4b via enp0s3
Jun 28 12:43:40 sysresccd dhcpd[3047]: DHCPACK on 192.168.1.101 to 08:00:27:d1:26:4b via enp0s3
```

Figura 27. Log servicio dhcpd en servidor DHCP

6.2 Prueba de servicio tftpd/pxe

En este apartado se comprueba el correcto funcionamiento del servicio PXE corriendo sobre System Rescue CD en la parte del servidor. Como se comentó en el anterior apartado, System Rescue CD proporciona la herramienta necesaria para levantar el servicio PXE, [figura 6](#) del [capítulo 3.2](#).

Se comprueba que el servicio tftpd se encuentra activo en el servidor, [figura 28](#). Este servicio será el encargado de transferir a los clientes pxe los archivos necesarios indicados anteriormente en los [capítulos 3 y 3.1](#).

`$/etc/init.d/in.tftpd status`

```
root@sysresccd /root % /etc/init.d/in.tftpd status
* status: started
```

Figura 28. Estado servicio tftpd

En la parte del cliente, los clientes pxe cargan correctamente el kernel y la imagen de disco inicial (initrd) para finalmente descargar el sistema de ficheros comprimido de System Rescue CD. En la [figura 29](#) se visualiza como el cliente1 tras la descarga y ejecución del kernel e initrd está cargando el sistema de ficheros (archivo “syscd.dat”) de unos 494 MB. La descarga de dicho fichero por TFTP tarda unos pocos segundos.

```

Cliente1 [Corriendo] - Oracle VM VirtualBox
>> Activating dmraid (fake hardware raid)...
>> Loading MD modules for software raid...
>> Starting mdadm (linux software raid)
mdadm: No arrays found in config file or automatically
>> Setting up the Logical Volume Manager
    Reading all physical volumes. This may take a while...
>> Performing the network configuration...
Here are the ethernet interfaces found on your system:
* eth0: 08:00:27:0A:F8:16
--- eth0: link=link-ok, mac=08:00:27:0A:F8:16
>> Attempting to get a DHCP address on eth0...
udhcpd (v1.18.2) started
Sending discover...
Sending select for 192.168.1.101...
Lease of 192.168.1.101 obtained, lease time 21573
set new default route: 192.168.1.254
busybox udhcpd -n -i eth0 --> 0
>> Downloading http://192.168.1.5/sysrcd.dat ...
Connecting to 192.168.1.5 (192.168.1.5:80)
null 100% |*****| 45 --:--:-- ETA
Connecting to 192.168.1.5 (192.168.1.5:80)
sysrcd.md5 100% |*****| 45 --:--:-- ETA
Connecting to 192.168.1.5 (192.168.1.5:80)
sysrcd.dat 100% |*****| 493M 00:00:00 ETA

```

Figura 29. Carga sistema de ficheros SRCD por TFTP

6.3

Modificación de imagen de System Rescue CD v5.3

Para el desarrollo del proyecto, se ha modificado la imagen ISO de System Rescue CD para disponer dentro del propio LiveCD de los ficheros necesarios (pxelinux.cfg y autorun0) para el funcionamiento de la herramienta. Por otra parte, se ha desarrollado otro script llamado “script_provisionado”. Este script, [figura 31](#), ejecutado desde System Rescue CD en el equipo servidor se encarga de configurar la red y copiar los ficheros a sus respectivas ubicaciones.

Para modificar imágenes ISO (o añadir ficheros), existen herramientas de terceros o se puede realizar directamente por la línea de comandos. El software elegido para modificar la imagen ha sido “Magic ISO” ejecutado sobre un equipo Windows. Este software permite abrir una imagen ISO, modificar (o añadir) ficheros y reconstruir la imagen ISO con los cambios aplicados.

Una vez arrancado el LiveCD de System Rescue CD se comprueba que en la ruta “/livemnt/boot/” se tiene acceso a los ficheros de la imagen ISO y a nuestros ficheros que contienen la herramienta desarrollada ([figura 30](#)). Se ejecuta el script “script_provisionado” de la imagen ISO y compruebo que este, copia los ficheros correspondientes, inicia el servicio PXE y lanza la herramienta desarrollada.

\$cd /livemnt/boot

```

root@sysresccd / % cd /livemnt/boot
root@sysresccd /livemnt/boot % ls
boot      efi      isolinux  readme.txt  sysrcd.md5  usb_inst.sh  version
bootdisk  Files    ntpasswd  sysrcd.dat  usb_inst    usbstick.htm
root@sysresccd /livemnt/boot % cd Files
root@sysresccd /livemnt/boot/Files % ls
Howto_Script.txt  ping_uc  pxelinux.cfg
root@sysresccd /livemnt/boot/Files % cd ping_uc
root@sysresccd /livemnt/boot/Files/ping_uc % ls
autorun0  script_provisionado

```

Figura 30. Ficheros imagen LiveCD SRCD

```

echo "Eres el servidor"

ip_servidor=192.168.1.5
ifconfig enp0s3 $ip_servidor
route add default gw 192.168.1.1
echo "Red configurada:"
ifconfig -a enp0s3 | grep "inet 1*"
ip route show | grep default
read -p "Presiona enter si es correcto, Ctrl^C para cancelar"

/etc/init.d/pxebootdrv start
cp -f /livemnt/boot/Files/pxelinux.cfg/default /tftpboot/pxelinux.cfg/
mkdir /tftpboot/ping_uc
cp -f /livemnt/boot/Files/ping_uc/autorun0 /tftpboot/ping_uc/
chmod 700 /tftpboot/ping_uc/autorun0
/tftpboot/ping_uc/autorun0

```

Figura 31. Código script "script_provisionado"

6.4 Pruebas herramienta en entorno virtual

En este apartado se mostrará la ejecución de la herramienta desarrollada tanto en la parte del servidor como en la parte de los clientes. El entorno virtual en el que he probado principalmente la herramienta es el nombrado en el [capítulo 5](#).

En este apartado, es preciso mencionar que en el script "autorun0" que despliega la herramienta se ha añadido una opción "test" ([figura 32](#)). Estableciendo esta opción en "yes" hace que tanto el servidor como los clientes muestren la secuencia de comandos que ejecutarían en lugar de ejecutarlos. Esta opción ha permitido un debugging y análisis del script de clonado de las opciones y particiones de disco.

```

...
# test_mode
# If option mark in yes, server and clients would show script but wouldn't run it
test=no
...
if [ "$test" = "yes" ]; then
    echo Test mode: Showing slave script
    cat $slave_script
else
    echo executing slave script...
    . $slave_script
    echo finished slave script...
...
if [ "$test" = "yes" ]; then

```

```

        echo Test mode: Showing slave script
        cat $master_script
    else
        echo executing master script...
        . $master_script
        echo finished master script...
    fi
...

```

Figura 32. Código “test” mode

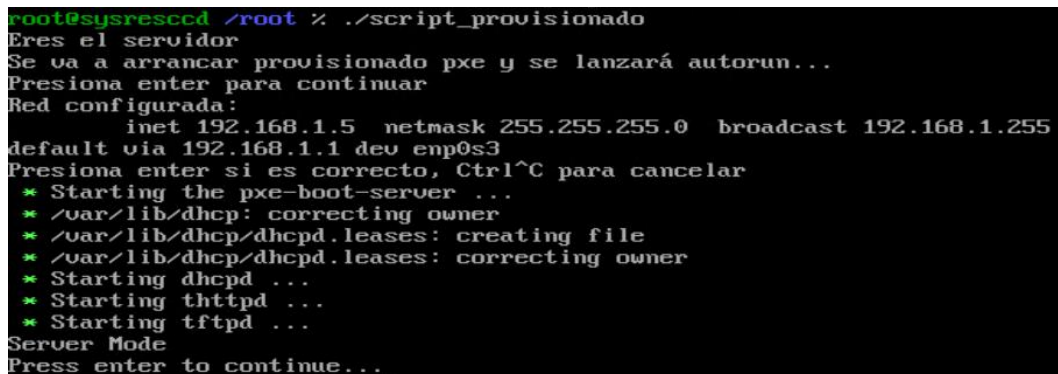
6.4.1 Pruebas desde parte servidor

Se arranca el equipo que actuará como servidor desde la imagen ISO de System Rescue CD modificada en el [capítulo 6.3](#). Como se puede ver en la [figura 33](#), una vez con SRCD arrancado, utilizo “script_provisionado” ([figura 31](#)) para configurar la red y lanzar la herramienta.

```

$cp /livemnt/boot/Files/ping_uc/script_provisionado .
$chmod 700 script_provisionado
$./script_provisionado

```



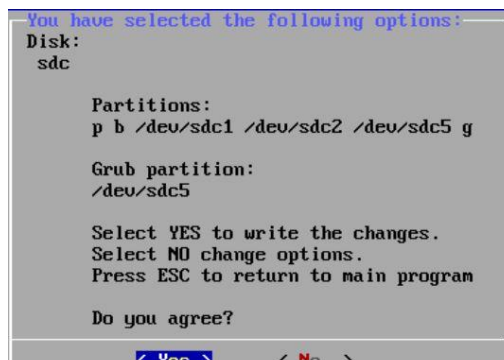
```

root@sysresccd /root % ./script_provisionado
Eres el servidor
Se va a arrancar provisionado pxe y se lanzará autorun...
Presiona enter para continuar
Red configurada:
    inet 192.168.1.5 netmask 255.255.255.0 broadcast 192.168.1.255
default via 192.168.1.1 dev enp0s3
Presiona enter si es correcto, Ctrl^C para cancelar
* Starting the pxe-boot-server ...
* /var/lib/dhcp: correcting owner
* /var/lib/dhcp/dhcpd.leases: creating file
* /var/lib/dhcp/dhcpd.leases: correcting owner
* Starting dhcpd ...
* Starting tftpd ...
* Starting tftpd ...
Server Mode
Press enter to continue..._

```

Figura 33. Ejecución “script_provisionado”

Al presionar la tecla “Enter” se ejecuta la parte de la herramienta correspondiente al servidor. Se ejecuta la interfaz gráfica de usuario ([capítulo 5.2](#)) donde el usuario elige el disco duro objetivo y las opciones y particiones a clonar hacia los clientes. A modo de prueba se va a clonar el disco duro completo, por lo que se eligen todas las opciones y particiones necesarias:



```

You have selected the following options:
Disk:
    sdc

Partitions:
    p b /dev/sdc1 /dev/sdc2 /dev/sdc5 g

Grub partition:
    /dev/sdc5

Select YES to write the changes.
Select NO change options.
Press ESC to return to main program

Do you agree?
< Yes > < No >

```

Figura 34. Confirmación opciones prueba

Como se observa en la [figura 35](#), una vez confirmadas las opciones y particiones de disco a clonar, el servidor realiza un análisis de las opciones elegidas y se queda a la espera de recibir conexiones de clientes para comenzar la transferencia.

```
target device: /dev/sdc
analysing target device...
found partition table on /dev/sdc
found master boot record on /dev/sdc
found NTFS partition on /dev/sdc1 (type 7)
estimated used space is 33000000 bytes
found NTFS partition on /dev/sdc2 (type 7)
estimated used space is 12325000000 bytes
found linux partition on /dev/sdc5 (type 83)
found linux partition on /dev/sdc5 (type 83) with filesystem ext4
estimated used space is 4980842496 bytes
will install grub from /dev/sdc5
sending slave script to slaves...
sending slave script to slaves...
Udp-sender 20120424
Using full duplex mode
Using mcast address 232.168.1.5
UDP sender for (stdin) at 192.168.1.5 on enp0s3
Broadcasting control to 192.168.1.255
```

Figura 35. Ejecución herramienta en servidor

En este punto se quiere recalcar que en el protocolo UDP-Sender existe la opción “-min-wait”. Esta opción permite configurar el tiempo que espera el servidor desde que recibe la primera petición de un cliente. De esta forma, se puede establecer el tiempo que necesitamos para arrancar todas las máquinas clientes por red y dejarlas preparadas para recibir la información desde el equipo servidor.

En la parte del servidor se puede visualizar a modo de logging las comunicaciones y órdenes que se están llevando a cabo entre los clientes y el servidor. También se puede visualizar el progreso de las diferentes transferencias de datos. En la [figura 36](#) se muestra a modo de ejemplo una ventana de logging obtenida del servidor.


```

sending slave script to slaves...
Udp-sender 20120424
Using full duplex mode
Using mcast address 232.168.1.5
UDP sender for (stdin) at 192.168.1.5 on enp0s3
Broadcasting control to 192.168.1.255
New connection from 192.168.1.101 (#0) 00000009
Starting transfer: 00000009
bytes=      1 915 re-xmits=00000000 ( 0.0%) slice=0112 - 0
Transfer complete.
Disconnecting #0 (192.168.1.101)

executing master script...
sending partition table from /dev/sdc
Udp-sender 20120424
Using full duplex mode
Using mcast address 232.168.1.5
UDP sender for (stdin) at 192.168.1.5 on enp0s3
Broadcasting control to 192.168.1.255
New connection from 192.168.1.101 (#0) 00000009
Starting transfer: 00000009
bytes=      365 re-xmits=00000000 ( 0.0%) slice=0112 - 0
Transfer complete.
Disconnecting #0 (192.168.1.101)

sending master boot record from /dev/sdc
sending master boot record from /dev/sdc
Udp-sender 20120424
Using full duplex mode
Using mcast address 232.168.1.5
1+0 records in
1+0 records out
UDP sender for (stdin) at 512 bytes copied, 0.00535174 s, 95.7 kB/s192.168.1.5
on enp0s3
Broadcasting control to 192.168.1.255
New connection from 192.168.1.101 (#0) 00000009

```

Figura 36. Logging servidor

6.4.2 Pruebas desde clientes

Una vez preparado el servidor, se arrancan por red los equipos clientes y se espera unos segundos a que arranquen el LiveCD de System Rescue CD como se mostró en el [capítulo 6.2](#). Una vez cargado SRCD, el servicio Autorun ejecuta la herramienta desarrollada, ya que así se le indicó en la opción “ar_source” de la [figura 4](#).

En las siguientes figuras se puede ver como ambos clientes cargan System Rescue CD y ejecutan la herramienta desarrollada desde “autorun0”:

```

Cliente1 [Corriendo] - Oracle VM VirtualBox
* Activating swap devices ... [ ok ]
INIT: Entering runlevel: 3
* Starting dbus ... [ ok ]
* Checking your configfile (/etc/syslog-ng/syslog-ng.conf) ... [ ok ]
* /var/lib/syslog-ng: correcting mode
* Starting syslog-ng ... [ ok ]
* Starting consolekit ... [ ok ]
* Checking for firmware to unpack ... [ ok ]
* Starting hsetup ... [ ok ]
* Auto-scrambling root password for security ... [ ok ]
* Starting autoconfig ...
* Hardware detection started ...
* Detected 1 Intel(R) Core(TM) i7-4700MQ CPU @ 2.40GHz CPU(s) @ 2394MHz [ ok ]
* Not Loading APM Bios support ...
* ACPI power management functions enabled ... [ ok ]
* PCMCIA disabled ...
* Skipping ALSA detection ...
* VideoCard: ImmoTek Systemberatung GmbH VirtualBox Graphics Adapter [ ok ]
* Starting NetworkManager ... [ ok ]
Connecting..... is [online]
ssh-keygen: generating new host keys: RSA DSA ECDSA ED25519
* Starting sshd ... [ ok ]
* Loading fonts and keymaps ...
* Caching service dependencies ... [ ok ]
* Performing the SystemRescueCd specific initializations ...
* Performing network configuration ...
* Starting SystemRescueCd autorun scripts ...
Initializing autorun...

=====executing autorun0=====
initrd=initramfs.gz BOOT_IMAGE=rescue64 scandelay=5 netboot=http://192.168.1.5/sysrcd.dat setknmap=es
ar_source=http://192.168.1.5/ping_uc/
Client Mode
Cloning will start with options chosen in server side
Press enter to continue...

```

Figura 37. Ejecución autorun0 en cliente1

```

Cliente2 [Corriendo] - Oracle VM VirtualBox
* Activating swap devices ... [ ok ]
INIT: Entering runlevel: 3
* Starting dbus ... [ ok ]
* Checking your configfile (/etc/syslog-ng/syslog-ng.conf) ... [ ok ]
* /var/lib/syslog-ng: correcting mode
* Starting syslog-ng ... [ ok ]
* Starting consolekit ... [ ok ]
* Checking for firmware to unpack ... [ ok ]
* Starting hwsuap ... [ ok ]
* Auto-scrambling root password for security ... [ ok ]
* Starting autoconfig ...
* Hardware detection started ...
* Detected 1 Intel(R) Core(TM) i7-4700MQ CPU @ 2.40GHz CPU(s) @ 2394MHz [ ok ]
* Not Loading APM Bios support ...
* ACPI power management functions enabled ... [ ok ]
* PCMCIA disabled ...
* Skipping ALSA detection ...
* VideoCard: InnoTek Systemberatung GmbH VirtualBox Graphics Adapter [ ok ]
* Starting NetworkManager ... [ ok ]
Connecting..... is [online]
ssh-keygen: generating new host keys: RSA DSA ECDSA ED25519
* Starting sshd ... [ ok ]
* Loading fonts and keymaps ...
* Caching service dependencies ... [ ok ]
* Performing the SystemRescueCd specific initializations ... [ ok ]
* Performing network configuration ...
* Starting SystemRescueCd autorun scripts ...
Initializing autorun...

=====executing autorun0=====
initrd=initramfs BOOT_IMAGE=rescue64 scandelay=5 netboot=http://192.168.1.5/sysrescd.dat setknmap=es
ar_source=http://192.168.1.5/ping_uc/
Client Mode
Cloning will start with options chosen in server side
Press enter to continue...

```

Figura 38. Ejecución autorun0 en cliente2

Se presiona “enter” para continuar (todos estos mensajes de prueba se podrían eliminar) y tras permanecer a la espera el tiempo establecido en la parte del servidor, los clientes abren la comunicación con el servidor.

```

CLIENT
Press enter to continue...
target device: /dev/
Udp-receiver 20120424
UDP receiver for (stdout) at 192.168.1.101 on eth0
received message, cap=00000009
Connected as #0 to 192.168.1.5
Listening to multicast on 232.168.1.5

```

Figura 39. Comunicación cliente1

```

CLIENT
Press enter to continue...
target device: /dev/
Udp-receiver 20120424
UDP receiver for (stdout) at 192.168.1.102 on eth0
received message, cap=00000009
Connected as #1 to 192.168.1.5
Listening to multicast on 232.168.1.5

```

Figura 40. Comunicación cliente2

Tras exceder el tiempo de espera el servidor comienza a enviar información a los clientes conectados. Los clientes muestran en todo momento lo que están realizando y el progreso de las transferencias. Por ejemplo:

```

receiving partition table for /dev/sdc
Udp-receiver 20120424
UDP receiver for (stdout) at 192.168.1.101 on eth0
received message, cap=00000009

/dev/sda1: Created a new partition 1 of type 'HPFS/NTFS/exFAT' and of size 100 MiB.
/dev/sda2: Created a new partition 2 of type 'HPFS/NTFS/exFAT' and of size 35 GiB.
Partition #2 contains a ext3 signature.
/dev/sda3: Created a new partition 3 of type 'Extended' and of size 4.9 GiB.
/dev/sda4: Created a new partition 5 of type 'Linux' and of size 2 GiB.
Partition #5 contains a ext4 signature.
/dev/sda6: Created a new partition 6 of type 'Linux swap / Solaris' and of size 976 MiB.
Partition #6 contains a swap signature.
/dev/sda7: Created a new partition 7 of type 'Linux' and of size 2 GiB.
Partition #7 contains a ext4 signature.
bytes= 425 ( 0.00 Mbps)
Transfer complete.
/dev/sda8: Done.

New situation:
Disklabel type: dos
Disk identifier: 0x8f3e2aaf

Device Boot Start End Sectors Size Id Type
/dev/sda1 * 2048 206847 204800 100M 7 HPFS/NTFS/exFAT
/dev/sda2 206848 73644031 73437184 35G 7 HPFS/NTFS/exFAT
/dev/sda3 73646078 83884031 10237954 4.9G 5 Extended
/dev/sda5 73646080 77692927 4046848 2G 83 Linux
/dev/sda6 77694976 79693823 1998848 976M 82 Linux swap / Solaris
/dev/sda7 79695872 83884031 4188160 2G 83 Linux

The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.

```

Figura 41. Transferencia tabla de particiones


```

receiving ntfs filesystem for /dev/sdc1
Udp-receiver 20120424
UDP receiver for (stdout) at 192.168.1.101 on eth0
received message, cap=00000009
Connected as #0 to 192.168.1.5
Listening to multicast on 232.168.1.5
ntfscione v2017.3.23 (libntfs-3g)
Ntfscione image version: 10.1bps)
Cluster size      : 4096 bytes
Image volume size  : 104853504 bytes (105 MB)
Image device size  : 104857600 bytes
Space in use       : 33 MB (30.7%)
Offset to image data : 56 (0x38) bytes
Restoring NTFS from image ...
100.00 percent completed08 Mbps)
Syncing ...
bytes= 32 235 315 (143.21 Mbps)
Transfer complete.
receiving ntfs filesystem for /dev/sdc2
ntfscione v2017.3.23 (libntfs-3g)
Udp-receiver 20120424
UDP receiver for (stdout) at 192.168.1.101 on eth0
received message, cap=00000009
Connected as #0 to 192.168.1.5
Listening to multicast on 232.168.1.5
Ntfscione image version: 10.1bps)
Cluster size      : 4096 bytes
Image volume size  : 26065498112 bytes (26066 MB)
Image device size  : 26065502208 bytes
Space in use       : 12325 MB (47.3%)
Offset to image data : 56 (0x38) bytes
Restoring NTFS from image ...
_ 3.19 percent completed64 Mbps)

```

Figura 42. Transferencia partición NTFS

```

creating ext4 filesystem on /dev/sdc5
mke2fs 1.43.9 (8-Feb-2018)
Creating filesystem with 3964416 4k blocks and 991232 inodes
Filesystem UUID: ce565dd1-5128-4d71-a4c2-771743ff60c8
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done

receiving filesystem for /dev/sdc5
Udp-receiver 20120424
UDP receiver for (stdout) at 192.168.1.101 on eth0
received message, cap=00000009
Connected as #0 to 192.168.1.5
Listening to multicast on 232.168.1.5
bytes= 3 317 038K (213.03 Mbps)

```

Figura 43. Transferencia partición Linux

En la [figura 41](#) se observa cómo se transfiere y se escribe la tabla de particiones del disco duro objetivo del servidor al cliente. En la [figura 42](#) se transfieren las dos particiones NTFS pudiendo ver el progreso. Por último, en la [figura 43](#) se crea el sistema de ficheros con formato ext4 para después volcar los datos desde el servidor.

Una vez clonado el disco duro del servidor a los dos clientes, estos pueden arrancar desde su disco duro interno para cargar el Grub (reinstalado) y elegir el sistema operativo deseado, Windows o Debian en este caso.

7. Conclusiones y trabajos futuros

7.1 Caso práctico de uso

Una empresa de gran envergadura con unos 900 empleados y unos 450 ordenadores de 4 tipos. SEG Automotive S.A Treto:

- Portátiles de oficinistas, son los ordenadores portátiles usados por los administrativos (contables, almacén, recursos humanos...). Deben ser equipos portátiles puesto que se deben poder mover dentro y fuera de la fábrica. Marca y modelo: Lenovo thinkPad 470.
- Portátiles de jefes de departamentos. Son los ordenadores portátiles de los jefes de grupo y directores de fábrica. Marca/modelo: HP 840 G4.
- Ordenadores de sobremesa oficinistas. Son ordenadores para realizar las mismas tareas que los primeros oficinistas pero que debido a su puesto no es necesario moverlos. Marca/modelo: HP EliteDesk 2019.
- Ordenadores de sobremesa industriales. Son ordenadores especialmente diseñados para entornos industriales con polvo, vibraciones y temperaturas altas. Marca/modelo: Wonderware ThinClient.

Partiendo del supuesto de que la empresa acaba de abrir una nueva sede o que debe cambiar todos sus ordenadores por un cambio de contrato de renting. Los cambios del parque de equipos informáticos son habituales en grandes entidades que alquilan sus ordenadores a cambio de renovarlos cada 4-5 años.

Tras recibir los 450 ordenadores hay que preparar un ordenador de cada tipo (SO con sus actualizaciones, software corporativo, drivers del sistema...) → Estimando unas 20 horas de trabajo (5 horas por cada equipo a preparar).

Considerando que el desarrollo de la herramienta del proyecto hasta dejarla a punto y 100 % funcional ha sido de unas 220 horas (incluida investigación, implementación, pruebas y documentación).

Valorando un periodo de tiempo de 10 años, durante el cual se han hecho dos cambios del parque de equipos (1 cambio cada 5 años). Esto se traduce en un total de 900 ordenadores de 8 tipos/modelos diferentes.

Instalación manual	Instalación usando herramienta desarrollada
900 ordenadores * 5 horas = 4500 horas	220 horas desarrollo herramienta.
Con una jornada laboral de 8 horas: 563 días	5 horas por máquina * 8 máquinas (8 tipos) = 40 horas

	<p>Suponemos 4 horas (en el peor de los casos) para clonar cada lote de equipos de un tipo. Suponemos discos duros comunes de 500 GB de espacio y una red Full-Duplex Gigabit.</p> <p>220 horas + 40 + (8 * 4) = 292 horas, 37 días.</p>
--	---

Figura 42. Tabla comparativa beneficios

En este caso práctico, se ha considerado la clonación completa de discos duros de equipos de nueva instalación, pero también se puede emplear la herramienta como apoyo al mantenimiento y resolución de incidencias. Por ejemplo, a la hora de sustituir un disco duro por una unidad nueva (para aumentar la capacidad o sustitución de HDD por SSD) o para la recuperación de particiones de disco dañadas por el motivo que sea. De esta forma, se puede usar la herramienta desarrollada como aplicación de clonado de discos al estilo de Acronis o Clonezilla.

En el desarrollo de este caso práctico se ha tomado como referencia una gran empresa, pero la herramienta desarrollada es 100% extrapolable a administraciones públicas, colegios o a la propia Universidad para el despliegue de nuevos laboratorios. Cuantas más máquinas se clonen usando la herramienta, más provecho y tiempo se ganará al rentabilizar el tiempo invertido en el desarrollo de la herramienta.

7.2 Conclusiones

System Rescue es una distribución Linux para sistemas con arquitectura x86 o x64. Como cualquier Live-CD funciona sobre la memoria RAM del equipo sin necesidad de instalación o modificación del disco duro y destaca por ofrecer (si se desea) un escritorio no muy bonito, pero sí muy ligero. Cuenta con un conjunto de herramientas para la recuperación de datos y la reparación de sistemas dañados como GParted. La versión usada para el proyecto (v5.3) es una distribución GNU/Linux basada en Gentoo y la última versión lanzada (v8.0, marzo 2021) ha pasado a estar basada en Arch Linux. El uso de esta distribución en este proyecto me ha permitido el desarrollo de una herramienta completamente portable en la que he podido automatizar las tareas que no requieren interacción con el usuario.

Con la elaboración de este proyecto se ha logrado desarrollar una herramienta funcional que cubre el objetivo inicial y satisface los requisitos nombrados en los [capítulos 4.2 y 4.3](#). La herramienta desarrollada puede ser utilizada en muy poco tiempo en cualquier escenario en el que contemos con una red LAN y en los que los equipos cumplan los requisitos hardware nombrado en el [capítulo 4.4](#). Como he podido comprobar en mi corta vida laboral, estos requisitos se cumplen en el 99 % de los equipos comerciales. Además, se ha desarrollado una herramienta gráfica sencilla de usar para que cualquier usuario con unos mínimos conocimientos de informática pueda usar la utilidad siguiendo unas mínimas instrucciones.

Con el desarrollo del proyecto me he demostrado a mí mismo que he podido completar los estudios que siempre he querido y que por fin podré desarrollar mi trayecto profesional con la formación que tan importante es tener.

Finalmente valoro los conocimientos y sentido crítico de investigación que he adquirido durante estos últimos años de estudio y trabajo. En los últimos 3 años, ejerciendo de técnico de soporte y sistemas he podido utilizar el servicio ejemplificado en este proyecto sin ser consciente de ello. He usado prácticamente a diario la instalación o clonado de sistemas por red (PXE), sufriendo en ocasiones algunos problemas que no era capaz de resolver al carecer de conocimientos suficientes o documentación alguna. Con el desarrollo del proyecto, dispongo de los conocimientos suficientes para desplegar o mantener un servidor de pxe además de disponer de una herramienta que puedo usar profesionalmente.

7.3 Trabajos futuros

Aunque la herramienta es funcional y se puede utilizar tal y como se ha descrito en este documento, existen algunos puntos de mejora o ampliación que se describirán a continuación.

- Tanto en el servidor como en la parte de los clientes, se podrían **incluir barras de progreso más sofisticadas**. Para mostrar el progreso de las transferencias he usado el comando “pv” (pipe viewer) que proporciona unas barras sencillas, pero poco vistosas visualmente. Por ejemplo:

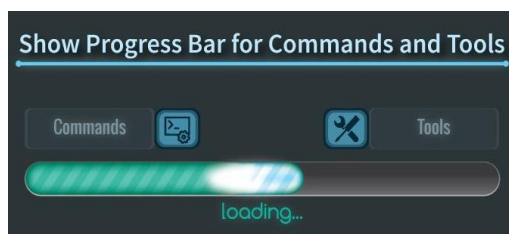


Figura 43. Ejemplo ventana de progreso [2](Compilar News, 2021)

Relacionado con esta mejora, se podría elaborar una ventana de progreso que mostrase el progreso global de las transferencias en lugar de mostrar las transferencias individualmente para cada opción.

- Elaborar una herramienta para el **filtrado y procesado del log de ejecución** del script. Dado que el log de ejecución se muestra directamente en la consola del servidor y clientes, se podría procesar esta información para mostrarla de una manera más amigable y fácil de interpretar.
- **Emplear LiveCDs más sofisticados y actualizados**. Se ha utilizado el LiveCD de SystemRescueCD como base para la ejecución de la herramienta, pero esta se puede ejecutar en cualquier distribución que proporcione una consola de Bash. Se ha elegido SRCD v5.3 ya que permite automatizar en gran medida la puesta en marcha del servicio PXE y los servicios asociados. Sin embargo, esta distribución podría presentar limitaciones en los últimos

equipos informáticos lanzados al mercado. Por ejemplo, se pueden presentar incompatibilidades con los controladores de las últimas tarjetas de red.

Por otra parte, aunque la imagen completa de System Rescue CD pesa unos 700 MB (un tamaño contenido para las velocidades de transferencias actuales) existen en el mercado otras distribuciones más ligeras.

Como alternativa a System Rescue CD se plantea la migración de la herramienta a otra distribución actual, moderna y ligera. Por ejemplo, se podría usar la distribución “**SliTaz**”.

SliTaz GNU/Linux es un sistema operativo del tipo Live-CD que además permite ser instalado en el disco duro ocupando tan solo 100 MB. El sistema de ficheros raíz ocupa sobre los 100 MB y la imagen ISO unos 45 MB, lo que hace que sea una distribución extremadamente ligera capaz de funcionar óptimamente con tan solo 128 MB de memoria RAM [11](SliTaz, s. f.).

La opción más interesante para este proyecto es que, aunque SliTaz contiene la mayoría de las herramientas Linux básicas, este puede ser completamente personalizado al poseer un repositorio con 3350 paquetes. La herramienta desarrollada en este proyecto podría ser integrada en la distribución SliTaz al contar esta con los paquetes necesarios [4](Desde Linux, 2015).

8. Referencias bibliográficas

1. About: Bash. (s. f.). About: Bash. DBpedia. Recuperado junio de 2021, de <https://es.dbpedia.org/page/Bash>
2. Compilar News. (2021, 11 marzo). Cómo mostrar la barra de progreso para los comandos de Linux. Recuperado junio de 2021, de <https://conpillar.es/como-mostrar-la-barra-de-progreso-para-los-comandos-de-linux/>
3. Debian Wiki. (2019). *DHCP_Server*. Recuperado enero de 2020, de https://wiki.debian.org/DHCP_Server
4. Desde Linux. (2015). Disponible SliTaz 4.0: Una distro para PCs con 128Mb de RAM. Recuperado julio de 2021, de <https://blog.desdelinux.net/disponible-slitaz-4-0-una-distro-para-pcs-con-128mb-de-ram-d/>
5. Desde Linux. (2019). *Sysadmin: El Arte de ser un Administrador de Sistemas y Servidores*. Recuperado abril de 2021, de <https://blog.desdelinux.net/sysadmin-arte-administrador-sistemas-servidores/>
6. Ionos. (2019, 4 febrero). TFTP (Trivial File Transfer Protocol) : un protocolo muy simple para la transferencia de archivos. Recuperado enero de 2020, de <https://www.ionos.es/digitalguide/servidores/know-how/tftp/>
7. Ionos. (2019, 30 julio). El DHCP y la configuración de redes. Recuperado enero de 2020, de <https://www.ionos.es/digitalguide/servidores/configuracion/que-es-el-dhcp-y-como-funciona/>
8. Linkfang. (2021, 12 febrero). Preboot Execution Environment. Recuperado febrero de 2020, de <https://es.linkfang.org/wiki/PXE>
9. Proceso de arranque de Linux. (2020). Programador clic. Recuperado diciembre de 2020, de <https://programmerclick.com/article/9584468795/>
10. Redigit. (2019). *¿Cómo ser un buen Administrador de Sistemas o Sysadmin?* Recuperado abril de 2021, de <https://blog.redigit.es/como-ser-un-buen-administrador-de-sistemas-o-sysadmin/>

11. SLiTaz. (s.f.). SLiTaz GNU/Linux. SLiTaz GNU/Linux. Recuperado junio de 2021, de <https://slitaz.org/es/about/>
12. Syslinux. (2020, 18 diciembre). PXELINUX. Recuperado febrero de 2020, de <https://wiki.syslinux.org/wiki/index.php?title=PXELINUX>
13. System Rescue. (s. f.). *System Rescue Homepage*. Recuperado marzo de 2020, de <https://www.system-rescue.org/>